ISSN 0542 9951

11
1995

NAJSTARSZE PISMO INFORMATYCZNE W POLSCE – UKAZUJE SIĘ OD 1965 ROKU

informatyka

Biblioteka Główna
P/18771/95

miesięcznik profesjonalistów

OS/2 WARP

Podczas spotkania z prasą i przedstawicielami firm współpracujących IBM ogłosił 23 października br. wprowadzenie do sprzedaży swego sztandarowego systemu operacyjnego OS/2 Warp w wersji polskiej. Prace nad spolszczeniem rozpoczęły się w lutym; zakończono je więc nader szybko, co budzi uznanie tym bardziej, że roboty było istotnie huk: spolszczenie jest bardzo głębokie, aż do tekstów pomocy i komunikatów uruchamianej pod Warpem sesji DOS. O jakości tej pracy nie możemy się jeszcze – z uwagi na zbyt krótki czas zapoznawania się z systemem – wypowiedzieć w pełni kompetentnie, ale pierwszy rzut oka daje wrażenie wielce pozytywne.

OS 2/Warp jest pierwszym w pełni 32-bitowym systemem operacyjnym dostępnym w wersji spolszczonej na rynku komputerów osobistych. Firmie IBM udało się zresztą wyprzedzić Microsoft, zarówno gdy chodzi o wersję angielską, jak i polską: premiera Windows '95 po polsku odbyła się w Warszawie w dwa dni później.

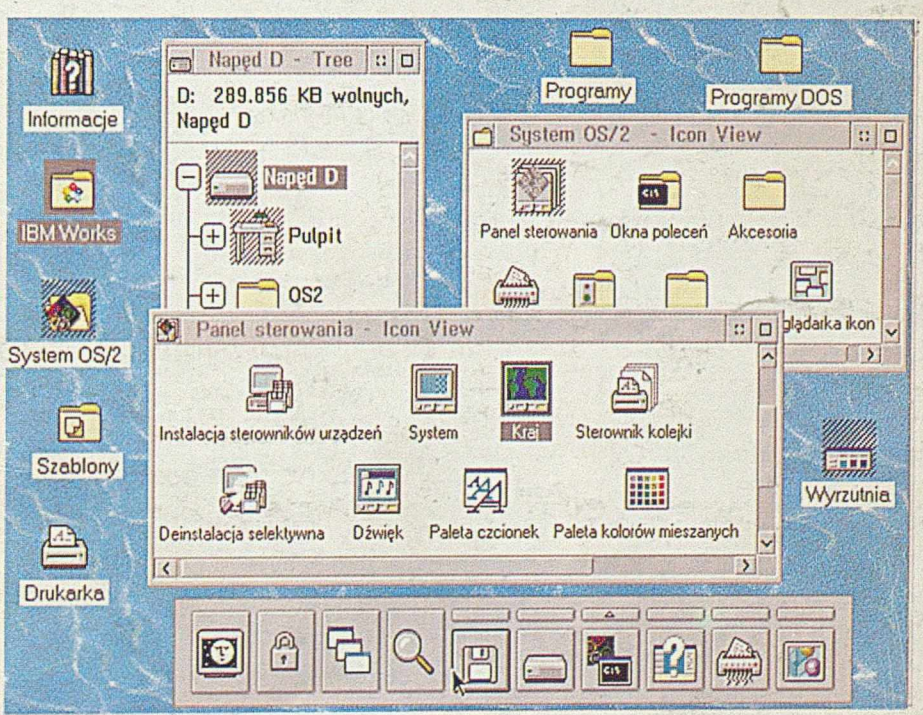
Sam system robi doskonałe wrażenie. Jest dopracowany w najmniejszych szczegółach i wyjątkowo przyjazny dla użytkownika, nie zabierając mu jednak – jeśli ten tego pragnie – przyjemności osobistego „ręcznego" konfigurowania sprzętu i dostarczając przy tym masy informacji. Zaawansowany i świadomy użytkownik będzie więc usatysfakcjonowany w pełni. System – zgodnie z informacją producenta – da się uruchomić nawet na komputerze klasy 386 z 4 MB RAM; należy to jednak rozumieć dosłownie. „Da się uruchomić" nie oznacza, że użytkownik będzie miał z tego rozkosz. Ale w końcu to aplikacje decydują o potrzebach; że zaś dzisiejsze programy „lubią" mieć do dyspozycji 16 MB RAM i procesor co najmniej 486 DX4, przeto nie bierzemy odpowiedzialności za użycie Warpa w gorszej konfiguracji...

Rynek otrzymał spolszczonego Warpa bez – niestety – tzw. *Bonus Pak*, czyli prostego, ale bardzo efektywnego pakietu zintegrowanego, zawierającego system zarządzania bazą danych, arkusz kalkulacyjny i procesor tekstu obok potężnych narzędzi do eksplorowania Internetu, prostej książki adresowo-telefonicznej, narzędzia do tworzenia raportów i wykresów i oprogramowania faksowego. Zestaw ten dostępny jest na razie tylko w jęz. angielskim, ale IBM zapowiedział, że wersja polska powinna być na rynku jeszcze w tym roku. *Bonus Pak* jest dostarczany przez producenta bezpłatnie jako część składowa systemu.

OS/2 Warp ze swoim wielce intuicyjnym interfejsem graficznym, który nie powinien w użyciu nastręczyć nikomu najmniejszych trudności, pozwala na pracę zarówno ze starymi programami do DOS-u, jak i z tymi do Windows; niestety, tylko do wersji 3.1 (dodajemy, że pracują one często wydajniej, niż w środowisku „oryginalnym"!). Kto zatem nabył już Windows '95, ten nie zdoła uruchomić sesji tego systemu spod Warpa ani odwrotnie; nie oznacza to jednak, że oba systemy nie mogą współistnieć w tym samym komputerze: wystarczy podzielić dysk na dwie partycje i w każdej zaimplementować inny system. Przy rozmiarach dzisiejszych dysków jest to całkiem możliwe, a może być i celowe, bowiem do Warpa jest coraz więcej bardzo ciekawych aplikacji – obecnie już ponad 2000. Przykładem może być choćby dostępny dla tego systemu Word Pro Lotusa, pocesor tekstu, znany dotychczas jako AmiPro 4. Do października sprzedano Warpa na świecie z górą 11 mln egzemplarzy. Zyskał on też ponad 50 ważnych nagród międzynarodowych. Przeszło 80 firm europejskich instaluje już Warpa na sprzedawanych lub produkowanych przez siebie komputerach.

Warto zwrócić uwagę, że jesteśmy obecnie świadkami wielkich przemian na rynku informatycznym: ostrej walki konkurencyjnej, koncentracji kapitału i wielu innych tego rodzaju zjawisk. Wyścig z czasem IBM i Microsoftu jest jednym z przejawów tej niezmiernie ciekawej sytuacji, z pewnością per saldo korzystnej dla klienta, bowiem owocującej nieuniknionymi obniżkami cen oraz poprawą parametrów jakościowych i sprzętu, i programów.

(BM)



WYDAWNICTWO SIGMA NOT

informatyka

nr 11 1995
Listopad
Rok wydania XXXI

Miesięcznik
ISSN 0542-9951
INDEKS 36124

REDAGUJE ZESPÓŁ:
mgr Bogdan **MIŚ**
(redaktor naczelny)
mgr Krystyna
KARWICKA-RYCHLEWICZ
(zastępca redaktora naczelnego)
mgr Teresa **JABŁOŃSKA**
(sekretarz redakcji)
Przemysław **BASZKIEWICZ**
(redaktor)
mgr Zdzisław **ŻURAKOWSKI**
(redaktor działu)
Alina **KLEPACZ** (sekretariat)

KOLEGIUM REDAKCYJNE:
prof. dr hab. Leonard BOLC,
mgr inż. Piotr FUGLEWICZ,
prof. dr hab. Jan GOLIŃSKI,
dr inż. Zenon KULPA,
doc. dr inż. Jan MULAWKA,
prof. dr hab. Wojciech OLEJNICZAK,
mgr inż. Jan RYŻKO,
dr Witold STANISZKIS,
dr inż. Jacek STOCHLAK,
prof. dr hab. Maciej STOLARSKI,
dr hab. Zdzisław SZYJEWSKI,
prof. dr hab. inż.
Ryszard TADEUSIEWICZ,
prof. dr hab. Jan WĘGLARZ

PRZEWODNICZĄCY
RADY PROGRAMOWEJ
prof. dr hab.
Juliusz Lech KULIKOWSKI

WYDAWCA:
Wydawnictwo Czasopism i Książek
Technicznych SIGMA NOT
Spółka z o.o.
ul. Ratuszowa 11
00-950 WARSZAWA
skrytka pocztowa 1004

Redakcja:
00-950 Warszawa,
ul. Ratuszowa 11, p. 628 i 644
skrytka pocztowa 1004
tel., fax: 619-11-61
tel.: 619-22-41 w. 159

Materiałów nie zamówionych redakcja nie zwraca

Autorzy artykułów proszeni są o przysyłanie tekstów na dyskietkach – w czystych kodach ASCII (Latin II, Mazovia) lub edytorach: Word 2, Word 6 – oraz odbitki na papierze.

Redakcja nie ingeruje w treść i formę ogłoszeń i innych materiałów reklamowych, w związku z tym nie ponosi za nie odpowiedzialności.

W sprawach ogłoszeń prosimy zwracać się bezpośrednio do Redakcji lub Działu Reklamy i Marketingu 00-950 Warszawa ul. Mazowiecka 12 telefon: 27-43-66 telefaks: 26-80-16

W numerze:

	Strona
OS/2 WARP – *Bogdan Miś*	I okł.
publikacje	
Programowanie w języku systemów eksperckich CLIPS(1). Podstawy składni języka CLIPS – *Tomasz Pytlak*	1
Perspektywy rozwoju kart płatniczych w Polsce (2). Problemy zabezpieczeń systemów kart płatniczych – *Jacek Biskupski*	2
Trudno jest pisać o sztucznej inteligencji (1). System ekspercki, akwizycja i reprezentacja wiedzy, wnioskowanie – *Halina Kwaśnicka*	10
Węzły wirtualne jako metoda zwiększenia odporności na błędy aplikacji działających w systemach rozproszonych – *Henryk Krawczyk, Piotr W. Umiński*	14
Systemy czasu rzeczywistego	
Metody analizy i specyfikacji wymagań. Podejście strukturalne do analizy i projektowania systemów z uwzględnieniem rozszerzeń dla systemów czasu rzeczywistego (1) – *Jan Kwiatkowski, Damian Misiek, Stanisław Szejko*	22
informacje	
Nowe multipleksery firmy RAD – **PB**	28
Usługi interakcyjne Microsoftu – **k**	28
ALM – ułatwienie dla programistów – **k**	29
HECTOR zmienia siedzibę i strategię – **k**	29

W najbliższych numerach:

- **Jacek Biskupski kończy cykl o perspektywach kart płatniczych w Polsce, omawiając atuty kart elektronicznych w rozwoju usług bankowych.**
- **Tomasz Pytlak kontynuuje cykl o języku CLIPS, charakteryzując w tej części strategie i mechanizmy wnioskowania.**
- **Halina Kwaśnicka podsumowuje uwagi na temat książki o sztucznej inteligencji.**
- **Jan Kwiatkowski, Damian Miślak i Stanisław Szejko kończą artykuł o podejściu strukturalnym do analizy i projektowania systemów.**

Warunki prenumeraty na 1996 r.

Zamówienia na prenumeratę czasopism wydawanych przez Wydawnictwo SIGMA-NOT można składać w dowolnym terminie. Mogą one obejmować dowolny okres czasu, tzn. dotyczyć dowolnej liczby kolejnych zeszytów każdego czasopisma.

Zamawiający może otrzymywać zaprenumerowany przez siebie tytuł począwszy od następnego miesiąca po dokonaniu wpłaty. Zamówienia na zeszyty sprzed daty otrzymania wpłaty będą realizowane w miarę możliwości – z posiadanych zapasów magazynowych.

Warunkiem przyjęcia i realizacji zamówienia jest otrzymanie z banku potwierdzenia dokonania wpłaty przez prenumeratora. Dokument wpłaty jest równoznaczny ze złożeniem zamówienia.

Wpłat na prenumeratę można dokonywać na ogólnie dostępnych blankietach w Urzędach Pocztowych (przekazy pieniężne) lub Bankach (polecenie przelewu), przekazując środki pod adres:
Wydawnictwo SIGMA-NOT Spółka z o.o.
Zakład Kolportażu
00-716 Warszawa, skr. poczt. 1004
konto:
PBK S.A. III 0/Warszawa nr 370015-1573-139-11

Wpłaty na prenumeratę przyjmują także wszystkie urzędy pocztowe nadawczo-odbiorcze oraz doręczyciele na terenie całego kraju

Na blankiecie wpłaty należy czytelnie podać nazwę zamawianego czasopisma, liczbę zamawianych egzemplarzy, okres prenumeraty oraz własny adres.
Na życzenie prenumeratora, zgłoszone np. telefonicznie, Zakład Kolportażu ul. Bartycka 20, 00-950 Warszawa, (telefony: 40-30-86, 40-35-89 oraz 40-00-21 wew. 249, 293, 299) wysyła specjalne blankiety zamówień wraz z aktualną listą tytułów i cennikiem czasopism.

Odbiorcy zagraniczni mogą otrzymywać czasopisma poprzez prenumeratę dewizową (wpłata dokonywana poza granicami Polski w dewizach, wg cennika dewizowego z cenami podanymi w dolarach amerykańskich) lub poprzez zamówioną w kraju prenumeratę ze zleceniem wysyłki za granicę (zamawiający podaje dokładny adres odbiorcy za granicą, dokonując równocześnie wpłaty w wysokości dwukrotnie wyższej niż cena normalnej prenumeraty krajowej).

Egzemplarze archiwalne (sprzedaż przelewowa lub za zaliczeniem pocztowym) można zamawiać pisemnie, kierując zamówienia pod adresem: Wydawnictwo SIGMA NOT, Spółka z o.o. Zakład Kolportażu, 00-716 Warszawa, ul. Bartycka 20, paw. B, tel. 40-37-31, natomiast za gotówkę można je nabyć w Klubie Prasy Technicznej w Warszawie ul. Mazowieckiej 12, tel. 26-80-17.

Istnieje możliwość zaprenumerowania 1 egz. czasopisma po cenie ulgowej przez indywidualnych członków stowarzyszeń naukowo-technicznych zrzeszonych w FSNT oraz przez uczniów zawodowych i studentów szkół wyższych. Blankiet wpłaty na prenumeratę ulgową musi być opatrzony na wszystkich odcinkach pieczęcią koła SNT lub szkoły.

W przypadku zmiany cen w okresie objętym prenumeratą Wydawnictwo zastrzega sobie prawo do wystąpienia o dopłatę różnicy cen oraz prawo do realizowania prenumeraty tylko w pełni opłaconej.
Cena jednego egzemplarza: normalna 5,00 zł (50 000 zł), ulgowa 3,75 zł (37 500 zł)
Wartość prenumeraty w zł:
Normalna: kwartalna 15,00 zł (150 000 zł), półroczna 30,00 zł (300 000 zł), roczna 60,00 zł (600 000 zł)
Ulgowa: kwartalna 11,25 zł (112 500 zł), półroczna 22,50 zł (225 000 zł), roczna 45,00 zł (450 000 zł)

Skład i druk: Drukarnia SIGMA NOT Sp. z o.o. z. 354/95

publikacje

Programowanie w języku systemów eksperckich CLIPS (1)

Podstawy składni języka CLIPS

POLITECHNIKA ŚLĄSKA Biblioteka Główna w Gliwicach
P18771/85

Tomasz Pytlak

Samodzielna Pracownia Sztucznej Inteligencji

Instytut Podstaw Elektroniki

Politechnika Warszawska

Język CLIPS jest typowym przedstawicielem języków systemów eksperckich. Został opracowany przez Sekcję Sztucznej Inteligencji w NASA. Nazwa CLIPS pochodzi od słów *C Language Integrated Production System*. Ze względu na wiele zalet oraz łatwość integracji ze środowiskiem języka C, język ten wypiera ostatnio inne narzędzia służące do realizacji systemów eksperckich. Wbudowane mechanizmy przetwarzania list, mechanizmy wnioskowania i samomodyfikacji stawiają CLIPS-a przed językami sztucznej inteligencji, takimi jak LISP czy Prolog. Język ten znajduje szerokie zastosowanie, głównie ze względu na rozbudowaną bibliotekę, łatwość programowania i niski koszt. CLIPS z powodzeniem został użyty do realizacji ponad 4000 systemów eksperckich m.in. w obronności, we wspomaganiu prac biurowych, w weryfikacji kontraktów na szczeblu rządowym, we wspomaganiu prac naukowych, w zarządzaniu, w przemyśle, w pracach wielu małych firm prywatnych. Język CLIPS jest reklamowany przez NASA jako najlepszy produkt umożliwiający realizację systemów eksperckich. Również w Polsce zostały opracowane pierwsze systemy eksperckie w tym języku.

Niniejsza praca otwiera krótki cykl artykułów poświęconych językowi CLIPS. Cykl ten został tak pomyślany, aby Czytelnik mógł opracować swój własny prosty system ekspercki już po przeczytaniu pierwszej części, a następnie dobudowywać do niego coraz bardziej skomplikowane struktury. W drugim artykule z tej serii Czytelnik pozna zasady sterowania wnioskowaniem w języku CLIPS poparte kilkoma przykładami. W trzecim zostaną wprowadzone elementy programowania obiektowego. Wreszcie, na zakończenie cyklu, przedstawimy przykłady zaawansowanych technik programowania i komunikacji systemu eksperckiego opracowanego w języku CLIPS, z programami opracowanymi w innych językach i środowiskach. Jednak podstawowym środowiskiem uruchomieniowym, pozwalającym napisać i przetestować system ekspercki jest środowisko windows/menu/mouse[1].

Elementy składni języka CLIPS

System ekspercki można uruchomić na jeden z trzech sposobów:

a) w trybie interakcyjnym,
b) korzystając z edytora,
c) dołączając gotowy produkt do programu nadrzędnego.

[1] Po wywołaniu programu na ekranie pojawia się okienkowy system podpowiedzi. Wyjściowa pozycja kursora znajduje się w okienku edycyjnym. Tutaj można wprowadzić konstuktory systemu eksperckiego. Jest to również podstawowe okno pozwalające na uruchomienie własnego systemu eksperckiego.

Mgr inż. Tomasz PYTLAK ukończył w 1986 r. Wydział Elektroniki Politechniki Waszawskiej, kierunek Aparatura Elektroniczna. Od trzech lat jest członkiem Samodzielnej Pracowni Sztucznej Inteligencji w Instytucie Podstaw Elektroniki PW. Jest autorem wielu opracowań i programów z dziedziny sztucznej inteligencji, systemów eksperckich i doradczych.

Zastosowaniom trzeciego ze sposobów poświęcimy jeden z dalszych artykułów cyklu.

Obecnie zajmijmy się uruchomieniem prostych funkcji i reguł w **trybie interakcyjnym**. Interakcyjny sposób pracy dotyczy raczej prostych implementacji. Datego też stosuje się go, na przykład, do kontroli pracy systemu eksperckiego, zapisanego wcześniej w pamięci. Przykład wywołania funkcji w trybie interakcyjnym może być następujący:

```
CLIPS>(+4 5)
9
CLIPS>
```

W tym wypadku po znaku zachęty wywołuje się funkcję dodawania, która jest zapisana w nawiasach. Należy zwrócić uwagę, że – jak każda funkcja arytmetyczna – jest ona **zapisana w postaci prefiksowej**. Jest to generalna zasada tworzenia oprogramowania w języku CLIPS. Użytkownik zechce sprawdzić następujące przykłady funkcji:

```
CLIPS>(eq 3 4)                    ;porównanie,
                                  ;nierówne: neq
FALSE
CLIPS>(> 5 3 1)                   ;inne często używane:
                                  ;<, <=, >=
TRUE
CLIPS>(create$ młot pilnik piła)  ;zwraca zmienną
                                  ;typu multiple[2)]
(młot pilnik piła)
CLIPS>(nth$ 3 (create$ a b c d e f))
c                                 ;zwraca element wskazany
                                  ;w zmiennej typu multiple
CLIPS>(member$ c (create$ a b c d e f))
3                                 ;zwraca numer elementu
                                  ;w zmiennej typu multiple
CLIPS>(str-cat „abc” „def”)       ;łączy argumenty
                                  ;w jeden łańcuch
abcdef
CLIPS>(+ 2 3.0 5)                 ;dodawanie
10.0
CLIPS>(– 12 3 4)                  ;odejmowanie
5
CLIPS>(* 3.1 4.7)                 ;mnożenie
14.57
CLIPS>(/ 24 3 4)                  ;dzielenie, dzielenie
                                  ;w zbiorze liczb
                                  ;naturalnych: div
2.0
CLIPS>(max 2 3 5 4 1)             ;maksimum, minimum:
                                  ;min, moduł: abs
5
CLIPS>
```

Dostępne są również funkcje jednoargumentowe, zwracające liczbę rzeczywistą, jak np. arccos: **acos**, arccos hiperboliczny: **acosh**, **cos**, pierwiastek kwadratowy: **sqrt**, logarytm dziesiętny: **log10**, np.

```
CLIPS>(round 3.6)                 ;zaokrąglenie
4
CLIPS>
```

Bardzo przydatne jest deklarowanie zmiennych globalnych. Sekwencja instrukcji, wykorzystująca zmienną globalną oraz funkcję przypisywania jej wartości, jest następująca:

```
CLIPS>(defglobal ?*x* = 5)        ;definicja zmiennej
                                  ;globalnej
CLIPS>?*x*
5
CLIPS>(bind ?*x* (** 3 2))        ;uaktualnienie wartości,
                                  ;potęgowanie: **
9
CLIPS>?*x*
9
CLIPS>
```

Instrukcja **bind** służy do przypisywania zmiennym wartości, a jej ogólna składnia wygląda następująco:

(**bind** <*zmienna*> <*wyrażenie*>)

Pełny zestaw instrukcji można znaleźć w CLIPS Reference Manual, dostępnym w zbiorach instalacyjnych.

Do zapamiętania struktury systemów eksperckich, gdzie wykorzystuje się sekwencje instrukcji, służy edytor. Wywoływany jest on po wybraniu opcji **File**:

Alt-F, Editor.

W tym miejscu warto zauważyć, że opcja **Editor** znajduje się w grupie innych, pozwalających na ładowanie tekstu programu do bufora, czy zapamiętywanie tekstu w pliku zewnętrznym. Opcja **Editor** pozwala na manipulacje tekstem programu, podobnie jak to jest w innych znanych edytorach. Po zapamiętaniu tekstu programu należy wrócić do środowiska CLIPS i załadować program w celu uruchomienia go. Służy do tego opcja:

Alt-F, Load Constructs

w wyniku której w okienku dialogowym pojawia się tekst programu poddany kompilacji. Na końcu tekstu znajdzie się informacja o poprawności tekstu, sygnalizowana napisem True w wypadku braku błędów, bądź False w wypadku błędów w tekście programu.

Podstawowa konstrukcja systemu eksperckiego w języku CLIPS

Program, napisany w języku CLPS, składa się z *konstruktorów*. W języku CLIPS dostępnych jest 11 konstruktorów: **defmodule, defrule, deffacts, deftemplate, defglobal, deffunction, defclass, definstances, defmessagehandler, defgeneric, defmethod.** Do napisania prostych systemów potrzebne są dwa: **deffacts** i **defrule.** O innych wspomnimy, opisując bardziej zaawansowane metody programowania. Generalną zasadą jest umieszczenie całego konstruktora w nawiasach okrągłych. Podobnie, jak w innych językach, tekst zapisany między znakiem; a końcem linii traktowany jest jako komentarz.

Konstruktor **deffacts** służy do określenia faktów opisujących bazę wiedzy:

(**deffacts** <*nazwa-grupy-faktów*>[<*komentarz*>]<RHS-*wzorzec*>)

Cały konstruktor jest ujęty w nawiasy. Napis **deffacts** należy do symboli zastrzeżonych i oznacza początek definicji grupy faktów o wspólnej nazwie, zawartej w drugim atomie konstruktora. Listę faktów umieszcza się w polu *RHS-wzorzec*. Komentarz nie jest obowiązkowy.

Jako przykład można podać następującą listę faktów zapamiętaną w pliku zewnętrznym przy wykorzystaniu poleceń edytora:

```
(deffacts przyczyny-pożaru
(temperatura bardzo wysoka)(dym bardzo gęsty)(wiatr bardzo silny))
```

Powyższa deklaracja określa pewne zjawiska atmosferyczne. Interpretacja tych zjawisk może doprowadzić

[2)] Takiej deklaracji używa się do definiowania zmiennych zawierających wiele elementów różnego typu.

do określenia przyczyn lub przesłanek związanych ze skutkiem ich działania, np. pożarem. Lista faktów może być dowolnie długa. Zbiór list określający bazę wiedzy może być dowolnie liczny. Fakty, zgodnie z deklaracją, umieszczane są na *stosie faktów*. Fakty, zadeklarowane zgodnie z instrukcją **deffacts,** są rekonstruowane po każdym poleceniu systemowym **reset** służącym do wyzerowania stanów wewnętrznych maszyny wnioskującej.

Za interpretacje faktów odpowiedzialny jest *zbiór reguł*. *Reguła* jest podstawowym konstruktorem języka CLIPS, odpowiedzialnym za reprezentację wiedzy. Konstruktor reguły jest następujący:

```
(defrule <nazwa-reguły> [<komentarz>]
        [<deklaracja>] ;specyfikacje reguły
        <warunek>      ;Left-Hand Side (LHS)-pole warunków
        =>
        <akcja>)       ;Right-Hand Side (RHS) –
                       ;lista wykonywanych akcji
```

Lewa strona reguły składa się z *warunków* (*conditional-elements* – **Ces**). Jeżeli warunki te są spełnione, to aktywowane są komendy z części RHS reguły. Separatorem między warunkami (Ces) a RHS jest znak = >. Przykładem mogą być reguły wykorzystujące wcześniej zadeklarowane fakty:

```
(defrule reguła-alarmowa-1 „To przykład prostej reguły"
        (temperatura bardzo wysoka)
        (wiatr bardzo silny)
        =>
        (printout t „Pożar rozprzestrzenia się")
        (assert (pożar rozprzestrzenia się)))
(defrule reguła-alarmowa-2
        (temperatura ~bardzo duża ;zaprzeczenie stwierdzenia: ~
        (deszcz dość duży)
        =>
        (printout t „Zmniejszona gotowość straży pożarnej")
        (assert (zmniejszona gotowość straży-pożarnej)))
```

W tekście przedstawiono inną formę deklaracji faktu za pomocą słowa **assert**. Deklaracja ta nie zezwala na rekonstrukcję faktu po poleceniu **reset**.

Zbiór reguł umieszczonych w pliku, określającym wiedzę systemu eksperckiego, pracując kolektywnie określa sposób rozwiązania problemu. Reguły powinny być zatem opracowane zgodnie z ogólnymi zasadami, ujętymi w formie następujących zaleceń:

- reguły bardzo szczegółowe powinny znajdować się na początku stosu reguł,
- reguły ogólne, oparte na schematach, na końcu stosu reguł,
- reguły mające taką samą postać (LHS lub RHS) powinny być połączone w jedną.

Powyższe konstruktory, zapamiętane w pliku zewnętrznym, można załadować do pamięci komputera. Preferowana sekwencja instrukcji jest następująca:

```
CLIPS>(clear)                    ;zerowanie pamięci
                                 ;komputera
CLIPS>(load „nazwa-pliku")       ;załadowanie
                                 ;konstruktorów do pamięci
CLIPS>(reset)                    ;zerowanie stanu maszyny
                                 ;i licznika poleceń
CLIPS>(run)                      ;uruchomienie systemu
CLIPS>Pożar rozprzestrzenia się  ;komunikat, który
                                 ;powinien pojawić się
                                 ;na ekranie
```

Konsekwencją *odpalenia* reguły *reguła-alarmowa-1* jest dopisanie do bazy wiedzy faktu
(pożar rozprzestrzenia się)

Reguła ta została odpalona, ponieważ konstruktorem **deffacts** wprowadziliśmy do pamięci komputera fakty zgodne z jej LHS.

Istnieją dwa polecenia, służące do przeglądania listy faktów i reguł:

```
CLIPS>(facts)
f-0  (initial-fact)
f-1  (temperatura bardzo wysoka)
f-2  (dym bardzo gęsty)
f-3  (wiatr bardzo silny)
f-4  (pożar rozprzestrzenia się)
For a total of 4 facts.
CLIPS>(rules)
rule-1  reguła-alarmowa-1
rule-2  reguła-alarmowa-2
CLIPS>
```

Polecenia systemowe można wypisywać na ekranie w nawiasach lub korzystać z edytora poleceń **Execution.** Oprócz wspomnianych, do wyboru są jeszcze: **Step,** pozwalająca na pracę krokową (w trybie interakcyjnym polecenie **run** *n*, gdzie *n* – liczba odpalonych poleceń); **Watch,** pozwalająca na śledzenie każdej operacji wykonanej w trakcie uruchomienia (analogicznie **watch**); **Clear**, kasująca wszystkie konstruktory z pamięci. Po wykonaniu tej instrukcji konieczne jest ponowne wprowadzenie konstruktorów do pamięci. Do pełnej kontroli pracy programu służą ponadto okienka deklarowane w opcji **Window.** Możliwe jest uaktywnienie np. okienka, odpowiedzialnego za prezentację faktów. Stan zawartości okienek aktualizuje się po wykonaniu akcji zgodnej z listą poleceń systemowych.

Numery faktów są zgodne z kolejnością ich deklaracji. *Fakt zerowy* (initial-fact) pojawia się po wykonaniu polecenia (**reset**) i jest konieczny do uruchomienia każdego systemu. W bazę wiedzy można ingerować, kasując wybrane fakty poleceniem (**retract** ?*adres-faktu*). Również kolejność reguł jest zgodna z deklaracją. Od kolejności umieszczenia reguł na stosie zależy kolejność ich odpalania. Ma to duże znaczenie, gdyż z kolei od kolejności odpalenia może zależeć sposób rozwiązania danego problemu. Strategiom działania systemu w wypadku reguł o tym samym priorytecie poświęcony będzie następny artykuł cyklu.

Reguły systemu mogą być odpalane w pętli. Kontrolę nad pracą systemu można zorganizować za pomocą polecenia **test**. Poniższe dwie reguły realizują system pracujący w pętli kontrolowanej instrukcją **test**:

```
defrule inicjuj
        (initial-fact)
        =>
        (printout t „Wprowadź liczbę pętli ?max:„)
        (bind ?max (read))                ;czyta liczbę z klawiatury
        (assert (stan-pętli 1 ?max)))     ;deklaracja faktu
                                          ;określającego stan pętli
(defrule druk-stanu-pętli
        ?loo <-(stan-pętli ?licznik ?max) ;?loo przyjmuje wartość
                                          ;adresu faktu w pamięci
        (test (<=?licznik ?max))          ;kryterium odpalenia
                                          ;reguły
        =>
        (printout t „Kolejny numer odpalenia reguły „ ?licznik crlf)
        (retract ?loo)                    ;polecenie skasowania
                                          ;faktu o adresie ?loo
        (assert(lstan-pętli=(+?licznik 1)?max))) ;deklaracja faktu
                                          ;z aktualnym stanem
                                          ;pętli
```

Po uruchomieniu systemu procedura druk-stanu-pętli zostanie odpalona *?max* razy. Warto zwrócić uwagę na formę realizacji pętli, Reguła druk-stanu-pętli zostaje odpalona pierwszy raz po wygenerowaniu faktu (stan-pętli 1 *?max*). Ten fakt jest kasowany przez tę regułę. Następnie, po wykonaniu dowolnej akcji, jest generowany nowy fakt z inkrementowaną wartością licznika pętli. Pojawienie się nowego dotychczas nie interpretowanego faktu spowoduje uaktywnienie procesu przeglądania reguł. Proces przeszukiwania spowoduje ponowne odpalenie reguły druk-stanu-pętli jako pierwszej zawierającej w polu LHS odpowiedni wzorzec. Funkcja test jest tu przykładowo użyta do badania kryterium stopu.

Przykłady prostych systemów eksperckich

Na zakończenie przedstawimy kilka przykładów użycia języka CLIPS wraz z opisami.

- Powtórzenie sekwencji akcji

```
(defrule powtórzenie
        (initial-fact)
        =>
        (printout t „Wprowadź daną" ?dana crlf);   przykład
                                                   ;danej
        bind ?dana (read))                         ;nadaje danej
                                                   ;wartość
                                                   ;z klawiatury
        (printout t „Wprowadzona dana wynosi „ ?dana crlf)
        (printout t „Czy chcesz wprowadzić kolejną daną? Y/N")
        (assert (powtórzenie =(read))))            ;generuje
                                                   ;fakt
                                                   ;określający
                                                   ;kontynuację
(defrule kontynuacja
        ?powt <-(powtórzenie Y/y)                  ;test
                                                   ;kontynuacji
                                                   ;z zastosowa-
                                                   ;niem opera-
                                                   ;tora or ?init
<-(initial-fact)
        =>
        (retract ?powt ?init)                      ;zerowanie
                                                   ;stosu
                                                   ;faktów
        (assert (initial-fact)))                   ;ustawienie
                                                   ;warunku
                                                   ;z pola LHS
                                                   ;reguły
                                                   ;powtórzenie
```

W tym wypadku w pętli pracują dwie reguły. Jedna wywołuje drugą do momentu, aż z klawiatury podamy polecenie wyjścia z pętli.

- Przykład kilku reguł, interpretujących światła na skrzyżowaniu;

```
(defrule idź
        (światło zielone)
        => (printout t „Możesz iść" crlf))
(defrule uwaga
        (światło żółte|zielone-migające)
        => (printout t „Uwaga" crlf))
(defrule stój
        (światło ~ green)
        => (printout t „Stój" crlf))
```

- Przykład reguły będącej uzupełnieniem powyższego systemu eksperckiego o możliwość kreowania nowych:

```
(defrule kreator
        (światło ?kolor)
        =>
        (bind ?lista-1 „(defrule nowa-reguła-") (bind ?lista-2 (read))
        (bind ?lista-3 „(światło„)              (bind ?lista-4 (kolor)
        (bind ?lista-5 „) => (printout t'„       (bind ?lista-6(read)) ;rodzaj akcji
        (bind ?lista-7„'crlf))")
        (bind ?lista(str-cat ?lista-1?lista-2 ?lista-3 ?lista-4 ?lista-5 ?lista-6 ?lista-7))
                                                ;skleja listy w jedną
        (build ?lista))                         ;buduje nową strukturę według
                                                ;deklaracji ?lista
```

W wyniku zadziałania tej reguły pod wpływem faktu nie zinterpretowanego przez wcześniejsze, zostanie utworzony i dołączony do systemu nowy konstruktor. Po wykonaniu instrukcji **reset** konstruktor ten zostanie dołączony do systemu jako nowa reguła. Przykład ten pokazuje, jak istotna jest kolejność ułożenia reguł na stosie. Umieszczenie reguły *kreator* na początku systemu spowoduje, że żadna reguła z przedstawionych wcześniej nie zostanie odpalona. Dlatego też konieczne jest umieszczenie jej na końcu jako reguły, uzupełniającej dany system. Każda reguła utworzona przez regułę *kreator* zostanie umieszczona na początku systemu. Przykład ten pokazuje możliwości CLIPS-a jako języka umożliwiającego samomodyfikację systemów w czasie rzeczywistym.

LITERATURA

[1] CLIPS Reference Manual. Vol 1: Basic Programming Guide, CLIPS Version 6.0, Software Technology Branch, Lyndon B. Johnson Space Center, 1995

[2] Ignizio J.P.: Introduction to expert system. McGraw-Hill Inc. New York, 1991. □

Laboratorium komputerowe Akademii Ekonomicznej w Poznaniu

Akademia Ekonomiczna w Poznaniu wzbogaciła się w tym roku akademickim o nowe laboratorium komputerowe, które pozwoli na wprowadzenie do programu zajęć z zakresu zintegrowanych systemów informacyjnych zarządzania. Laboratorium zostało wyposażone w oprogramowanie firmy SAP, umożliwiające projektowanie, wdrażanie i eksploatację w pełni zintegrowanych systemów zarządzania przedsiębiorstwami, bankami, towarzystwami ubezpieczeniowymi, jednostkami samorządu terytorialnego. Oprogramowanie to wykorzystują w zarządzaniu takie firmy, jak IBM, SIEMENS, Deutsche Bank.

Laboratorium umożliwia też wspomaganie modelowania systemów informacyjnych, co jest szczególnie przydatne przy wdrażaniu zintegrowanych systemów lub przy reorganizacji procesów gospodarczych. Można wówczas modelować nowy proces, porównać go z istniejącymi, wybrać korzystniejszy wariant. Narzędzia do modelowania – ARISToolSet – zostały opracowane przez Człowieka Roku Gospodarki Niemiec w 1993 r. – prof. W.A. Scheera.

Prof. dr Witold Abramowicz, kierownik Katedry Informatyki Ekonomicznej w AE w Poznaniu podkreśla, iż nowe laboratorium pozwoli na rozpoczęcie procesu integracji przedmiotów informatycznych z takimi przedmiotami, jak rachunkowość, finanse, zarządzanie, marketing i inne, a tym samym zmodernizuje kształcenie w zakresie tych nauk. **(k)**

Perspektywy rozwoju kart płatniczych w Polsce (2)

Problemy zabezpieczeń systemów kart płatniczych

Jacek Biskupski
Ekspert PKN ds. kart identyfikacyjnych
DIGITCARD - UNICARD Sp. z o. o.
Kraków

ARTYKUŁ REKOMENDOWANY

Warto zauważyć, że począwszy od pierwotnej idei kart, wprowadzonych przez *Flatbush National Bank*, zastosowana płytka o standardowych wymiarach miała spełniać rolę identyfikatora konta, służącego jedynie przyspieszeniu procedury naniesienia danych na skrócony wniosek o przyznanie kredytu. Nawet późniejsze (1969 r.) wprowadzenie przez system *MasterCard* paska magnetycznego i jego zapisu, miało charakter przede wszytkim informacyjny, przyspieszając w amerykańskim – w pełni skomputeryzowanym – systemie bankowym wprowadzenie informacji o numerze posiadanego konta. To ostatnie założenie było właściwie realizacją opracowanej przez organizacje międzybankowe strategii rozwoju systemów kartowych, zakładającej centralizację danych o kartach i ich posiadaczach w dużych systemach komputerowych.

Identyfikacyjna bankowa karta płatnicza

Używanie magnetycznych kart identyfikacyjnych (ang. *pointers*) miało (i ma) na celu jedynie uzyskanie dostępu do informacji zawartej w tych systemach, zwanych dalej systemami on-line [7]. Konsekwencją takiego podejścia było przyporządkowanie (początkowo) drugiej ścieżki magnetycznej dla aplikacji bankowych – co dawało początkowo możliwość zapisu na karcie jedynie 37 znaków numerycznych (tzn. identyfikatora wydawcy, numeru konta klienta, daty ważności karty, kodu kraju oraz trzycyfrowego kodu informacji dodatkowej).

Pierwotna koncepcja systemu scentralizowanego nie mogła jednak nadążyć za rozwijającym się rynkiem. Nadspodziewanie szybko wzrastająca liczba kart będących w obiegu (w USA od chwili wprowadzenia systemu do dnia dzisiejszego liczba kart w obiegu wzrosła 300-krotnie, a transakcji 30-krotnie) doprowadziła do pojawienia się problemów z transmisją i przetwarzaniem danych. Taka sytuacja zmusza organizacje zarządzające systemami kartowymi do ustanowienia kompromisów pomiędzy szybkością i jakością pracy systemów a ich bezpieczeństwem. Doprowadziło to do zaakceptowania procedur uproszczonych, dopuszczających zawarcie transakcji bez konsultacji z bankiem-wydawcą karty, a więc jedynie na podstawie zaufania do klienta, akceptanta, karty.

Efektem wprowadzonych kompromisów jest notowany w ciągu ostatnich pięciu lat dramatyczny wzrost liczby nadużyć dokonywanych za pomocą kart płatniczych. Analiza powstających nadużyć prowadzi do wniosku, że ponad 98% z nich powstaje w wyniku prowadzenia transakcji *off-line* tzn. *bez uzgodnienia transakcji z bankiem-wydawcą karty lub instytucją rozliczającą*. Problem nabiera szczególnego wymiaru w przypadku płatności drobnych, gdzie z ekonomicznego punktu widzenia traktowanie karty jako *pointera* w systemie pracującym *on-line* jest zbyt drogie. Doszło więc do sytuacji, gdzie pasywne narzędzie, zaprojektowane jako klucz do scentralizowanego systemu, zaczęło funkcjonować jako aktywny element – nie tyle generujący dane do przeprowadzenia transakcji, co właściwe transakcje, w dodatku bez kontaktu z bazą danych. Dopuszczenie do transakcji na stosunkowo niewielkie i bezpieczne kwoty w trybie off-line kartami, zaprojektowanymi do działań w trybie on-line jest głównym źródłem nadużyć w systemach kartowych. Podjęte próby zwiększenia przepustowości systemów i ich uszczelnień przez dobudowywanie kolejnych pakietów programów (jak brytyjski Fraud 2000) również nie dają spodziewanych efektów. Działania te mają na celu jedynie ograniczenie liczby nadużyć w systemach kartowych, które nie wytrzymały dynamiki rozwoju, bez zmiany ich zasadniczej scentralizowanej struktury.

Zmiana struktury, ze względu na ogromne koszty, będzie możliwa prawdopodobnie dopiero przy kolejnej zmianie technologii, na przykład przy przejściu na system kart elektronicznych.

Dodatkowym realnym problemem jest awaryjność rozległych scentralizowanych systemów autoryzacyjnych – jak np. wypadek „zawieszenia się" na ponad 30 godzin największego francuskiego systemu autoryzacji kart bankowych w czerwcu 1994 roku. Wypadki takie mogą

Rekomenduje:
Prof. dr hab. Andrzej Baborski
Rektor Akademii Ekonomicznej
Wrocław

powodować – i powodują – ogromne straty, wystarczy wziąć pod uwagę fakt, że ww. system przetwarza we Francji ponad dwa miliardy transakcji rocznie.

Efektem rynkowym wspomnianych problemów jest:

- Notowane w ostatnich latach spowolnienie rozwoju międzynarodowych produktów kartowych – szczególnie operujących w trybie off-line kart kredytowych.
- Wzrost zainteresowania systemami kart debetowych – najczęściej organizowanych jako możliwe do sprawnego zarządzania lokalne systemy on-line.
- Coraz poważniejsze rozważanie użycia kart elektronicznych jako rozwiązania łączącego zalety systemów on- i off-line, następnego etapu ewolucji systemów kartowych.

W toku analizy sytuacji naszego kraju nasuwa się więc pytanie: czy lokalne, np. ogólnonarodowe, systemy kart płatniczych powinny powstawać przy wykorzystaniu idei scentralizowanych baz danych, czy też odpowiednio zabezpieczonych systemów rozproszonych? Ogólna dostępność sprawdzonych i dopracowanych systemów, działających przy wykorzystaniu scentralizowanych baz, pomimo przytoczonych powyżej niebezpieczeństw, wydaje się bardzo pewnym rozwiązaniem dla systemu kart płatniczych w kraju wielkości Polski. Z rozwiązania tego skorzystali już np. Węgrzy. Z drugiej strony, technologia kart elektronicznych daje możliwość tworzenia systemów pracujących off-line o poziomie bezpieczeństwa dużo wyższym niż aktualnie stosowane (on-line) scentralizowane systemy kartowe, nie wymagając dodatkowo sprawnych łączy teletechnicznych.

Podsumowując wykazaną powyżej zmianę funkcji magnetycznej karty płatniczej od pasywnego *pointera* do aktywnego instrumentu, warto wskazać na konsekwencje takiego działania, szczególnie widoczne w problemach nadużyć.

Problemy nadużyć w systemach kart płatniczych

Nadużycie kartowe to oszustwo na szkodę banku, akceptanta lub związanej z systemem osoby fizycznej, polegające na wyłudzeniu pieniędzy lub towarów przez niezgodne z regulaminem użycie kart płatniczych. Pojęcie to zawęża się najczęściej, wyłączając „złe kredyty" udzielone przez używanie kart kredytowych.

We współczesnych systemach kartowych – z powodów wymienionych poprzednio – jest używne pojęcie „zarządzania nadużyciami", będące świadomym działaniem mającym na celu rozpoznanie, kontrolę i walkę o utrzymanie minimalnego poziomu nadużyć w systemach kart płatniczych. Takie rozwiązanie nabiera bardzo wymownego charakteru w porównaniu z walką wypowiedzianą grupom przestępczym, zajmującym się podrabianiem znaków pieniężnych. Wszelkie innowacje mające na celu podniesienie poziomu bezpieczeństwa systemu kartowego są tym trudniejsze, im bardziej jest rozległy zasięg działania systemu. Nie sposób, na przykład, dokonać z dnia na dzień zmiany metod zabezpieczenia kart czy procedur zabezpieczających w systemach kartowych działających w skali międzynarodowej. Powszechna akceptowalność kart płatniczych w wymianie międzynarodowej opiera się na stosowaniu opracowywanych przez organizacje kartowe „specyfikacji", wg których np. dokonuje się kontroli samej karty, sposobu prowadzenia transakcji, autoryzacji, rozliczeń itd. Choć teoretycznie dla systemu, takiego jak *MASTER CARD INT.* wymiana będących w użyciu kart byłaby tańsza niż jednoroczne straty systemu (tylko nadużyć z powodu podrabiania kart [3] to w praktyce jednak i tak nie rozwiązałoby to problemu. Właśnie z powodu powszechnej (międzynarodowej) akceptowalności wymiana taka w praktyce oznaczałaby jedynie dodanie nowych elementów zabezpieczenia – z zachowaniem wszystkich poprzednich – redukując na jakiś czas tylko te nadużycia, które powstają w wyniku stosowania kart podrobionych (por. tab. 2). Głównym źródłem strat w systemach kartowych są nadużycia spowodowane przestarzałością organizacyjną całego systemu i jego założeń, nie zaś wyłącznie samych kart.

Tabela 1. Analiza źródeł nadużyć w systemach kart płatniczych

Czy karta jest tą, którą wydał bank?	Czy okaziciel karty jest tym, komu ja wydał bank?	Czy transakcja rzeczywiście nastąpiła?	Najprawdopodobniejszy efekt w takiej sytuacji	% udziału w nadużyciach w systemach międzynarodowych
+	-	+	karta skradziona lub zgubiona	> 40%
-	-	+	karta skutecznie podrobiona	< 30%
+	-	+	karta nie dotarła do klienta	> 15%
+/- ?	+/- ?	+/- ?	klient nie przyznaje się do transakcji	> 10%
+	-	+	„skradziony" PIN (*on-line*)	< 5%
+	+	-	inne	< 10%

Źródło: opracowanie własne
Uwaga: podane udziały procentowe są orientacyjne i nie należy ich sumować

Problemy nadużyć w systemach kart płatniczych koncentrują się wokół odpowiedzi na trzy podstawowe pytania:

- Czy karta jest prawdziwa?
- Czy okaziciel karty jest tym, za kogo się podaje?
- Czy rozpatrywana transakcja nastąpiła?

Tylko pozytywna odpowiedź na wszystkie trzy pytania jednocześnie jest gwarancją rzetelności transakcji.

W tabeli 1 zestawiono przykładowe sytuacje, w których jeden tylko nieprawdziwy element powoduje powstanie nadużycia. Analiza tabeli prowadzi do wniosku, że większość nadużyć w międzynarodowych systemach kartowych powstaje w wyniku braku możliwości skutecznego sprawdzenia tożsamości okaziciela karty, a praktyka dodatkowo uczy, że wprowadzone zabezpieczenia zawodzą wobec ciągle zaskakującej wyobraźni przestępców.

Podstawowe zagadnienie dla tej grupy problemów to czy w ogóle i na ile pewnie można stwierdzić w punkcie dokonania transakcji, czy przedstawiona karta jest oryginalna, czy też jest to tylko falsyfikat? Obecnie stosowane metody zabezpieczenia kart przed podrobieniem zostały zaprojektowane dla systemu on-line, opracowanego 25 lat temu. Według danych przedstawionych w [9] można stwierdzić, że w systemach międzynarodowych karty skopiowane lub przerobione powodują ok. 29% wszystkich nadużyć. Metody zabezpieczeń przed procederem podrabiania kart – zwane dalej metodami CAM (ang. *Card Authentication Methods*), polegają na utrudnieniu możliwości podrobienia karty przez wprowadzenie podczas jej przygotowania odpowiednich zabezpieczeń.

Zabezpieczenia kart płatniczych przed nadużyciem

Stosowane obecnie zabezpieczenia kart płatniczych można podzielić na dwie podstawowe grupy:
- zabezpieczenia jawne,
- zabezpieczenia niejawne.

Wybrane zabezpieczenia jawne

Wzór graficzny – zabezpieczenia graficzne. Zabezpieczenie kart płatniczych przez swoistą, zastrzeżoną grafikę (charakterystyczne logo systemu) ma charakter bardziej umowny niż praktyczny. Faktem jest, iż wydrukowanie pojedynczych egzemplarzy kart płatniczych przy stosowanej tu technologii druku jest inwestycją większą niż zyski z nadużyć dokonanych na pojedynczych kartach, choć poziom stosowanych zabezpieczeń jest o kilka rzędów niższy niż w wypadku druku banknotów. Ogromna jednak liczba producentów kart plastikowych, działając raczej na umownych zasadach zastrzeżeń dotyczących wzorów graficznych, jest w stanie dostarczyć karty nie różniące się niczym od oryginałów. W praktyce zorganizowane grupy przestępcze przygotowują masowo niestety doskonale podrobione karty.

Hologram to bardzo cienka folia metaliczna, na którą techniką laserową naniesiono rysunek lub tekst, który w zależności od projektu może stwarzać wrażenie dwu- lub trójwymiarowego. Po raz pierwszy umieszczony na karcie plastikowej przez system *MasterCard Int.*, ze względu na wysoki koszt przygotowania matrycy oraz niewielką liczbę producentów w dalszym ciągu stanowi jedno z lepszych jawnych zabezpieczeń kart płatniczych. Jego wielką zaletą jest możliwość szybkiej, nie wymagającej dodatkowych przyrządów i prawie bezbłędnej, oceny obecności i jakości właściego hologramu przez akceptanta w miejscu transakcji – np. w sklepie czy na stacji benzynowej.

Personalizacja kart, to nadanie karcie cech unikatowych przez przyporządkowanie jej unikalnego napisu (wytłoczonego lub naniesionego w sposób termiczny) lub (i) zapisu ścieżek magnetycznych. Nanosi się w ten sposób numery, nazwiska, daty ważności karty, zdjęcia jej posiadacza. Personalizacja jest bardzo ważnym elementem bezpieczeństwa systemów kartowych, ponieważ od tego momentu karta jest związana z kontem klienta banku, a przestając być plastikowym blankietem, zaczyna być instrumentem obrotu bezgotówkowego. W przypadku personalizacji metodą tłoczenia (tzw. *Embossingu*) międzynarodowych kart kredytowych istnieją dodatkowo specjalne znaki personalizacyjne niedostępne pozabankowym nabywcom urządzeń, podwyższające bezpieczeństwo wystawianych kart. Ostatnim etapem personalizacji karty jest złożenie na specjalnie przygotowanej niezmazywalnej powierzchni karty podpisu przez osobę, której karta została wydana. Jest to bardzo ważny, a jednocześnie najczęściej atakowany element personalizacji, ponieważ podpis, w przypadku transakcji w trybie off-line, staje się w wielu wypadkach jedynym elementem kontroli tożsamości osoby posługującej się kartą.

Zabezpieczenia niejawne

Mikroprint polega na nadrukowaniu lub wykryciu wzoru niewidocznego okiem nieuzbrojonym. Odpowiednie oświetlenie bądź powiększenie nadruku czyni go widocznym. Mikroprinty służą badaniu autentyczności karty na poziomie badań specjalistycznych, choć w praktyce niektóre ich wersje mogą być dostrzeżone już za pomocą szkła powiększającego. Nie stanowią one jednak zabezpieczenia, do weryfikacji którego zobowiązuje się akceptantów kart.

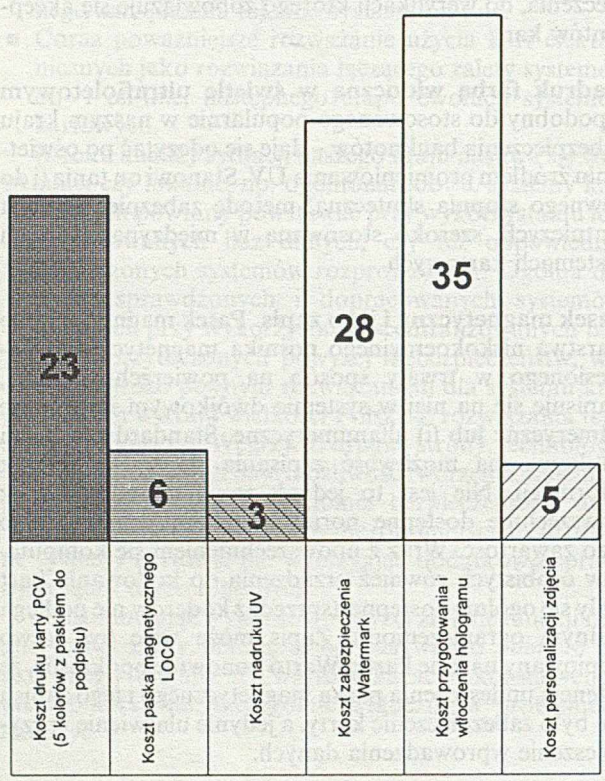
Nadruk farbą widoczną w świetle ultrafioletowym – podobny do stosowanego popularnie w naszym kraju zabezpieczenia banknotów – daje się odczytać po oświetleniu źródłem promieniowania UV. Stanowi on tanią (i do pewnego stopnia skuteczną) metodę zabezpieczeń kart płatniczych, szeroko stosowaną w międzynarodowych systemach kartowych.

Pasek magnetyczny i jego zapis. Pasek magnetyczny to warstwa niskokoercyjnego nośnika magnetycznego naniesionego w trwały sposób na powierzchnię karty. Zapisuje się na niej w systemie dwójkowym informacje numeryczne lub (i) alfanumeryczne. Standardowa karta bankowa ma możliwość zapisania do trzech ścieżek informacji. Nie jest to jednak zapis tajny, ponieważ powszechnie dostępne normy określają jednoznacznie jego zawartość. Wraz z upowszechnieniem się komputerów osobistych również urządzenia do kodowania kart stały się ogólnie dostępne, (sprzedaż koderów nie podlega żadnym ograniczeniom). Zapis może więc być łatwo kopiowany na inne karty. Warto ponownie podkreślić, że intencją umieszczenia paska magnetycznego i jego zapisu nie było zabezpieczenie karty, a jedynie ułatwienie i przyspieszenie wprowadzenia danych.

„Drugi klucz" – Watermark, SECO i MM. Wzrost nadużyć na kartach płatniczych obsługiwanych wyłącznie przez urządzenia elektroniczne (np. ATM, EFTPOS) spowodował w ostatnich latach gorączkowe poszukiwania „drugiego klucza", czyli dodatkowego zabezpieczenia karty magnetycznej obsługiwanej nie przez człowieka. Systemy zabezpieczeń (np. Watermark, kody SECO, czy MM) powodują wprowadzenie do karty dodatkowej, niewidocznej i nie dającej się usunąć informacji czytanej elektronicznie. Ma to na celu jednoznaczną identyfikację autentyczności karty przez urządzenie elektryczne. Skuteczność tych metod w znacznym stopniu zależy od sprawności samego zabezpieczenia oraz skali jego stosowania. Dla przykładu, szwedzki międzybankowy system kart gotówkowych *BANKOMAT* od połowy lat 80. jest w całości zabezpieczony dodatkową ścieżką magnetyczną systemu Watermark, co ma zasadniczy wpływ na niewielką liczbę nadużyć dokonywanych podrobionymi kartmi w tym kraju. Międzynarodowe systemy kartowe, jak np. VISA INT. również prowadzą niezależne testy tej technologii dla zabezpieczenia ATM i EFTPOS (np. w Northampton, GB).

Układ elektroniczny – karta hybrydowa. Wprowadzenie układu elektronicznego do standardowej karty o wymiarach ISO ma szansę zrewolucjonizować systemy bankowe. We Francji zdecydowano się już w 1992 r. wprowadzić układ elektroniczny w celu bezpiecznej kontroli kodu osobistego – PIN w transakcjach w trybie

off-line. W takim wypadku PIN – jest bezpiecznie zapisany w środku karty, a jego poprawne podanie jest sprawdzone wewnętrznie przez logikę karty. Daje to możliwość sprawdzenia PIN-u w miejscu transakcji, bez kosztów transmisji. Przeprowadzone próby dowiodły już skuteczności tej metody, a wprowadzony obowiązek stosowania układu elektronicznego w kartach płatniczych (CB) spowodował we Francji spadek poziomu nadużyć za pomocą kart płatniczych z 0,13 do 0,11% w latach 1990–1991, 0,11 do 0,07% w latach 1991–1992 i 0,07 do 0,05% w latach 1992–1993 [5]. Wymagało to jednak wymiany ponad 80% terminali EFTPOS zainstalowanych w punktach przyjmujących zapłatę kartami. Wymiana taka wydaje się jednak opłacalna, zważywszy, że ponad 90 transakcji we Francji jest aktualnie prowadzonych w trybie off-line.



Źródło: OFERTA UNICARD Kraków

Zestawienie kosztów wybranych metod zabezpieczeń kart płatniczych

Według organizacji *Europay International* przewidywane zyski z wprowadzenia zabezpieczeń produktów kartowych tą metodą szacuje się tylko w Europie na 2 miliardy ECU w ciągu następnych siedmiu lat [3].

Dla zobrazowania kosztów stosowania poszczególnych zabezpieczeń, w poniższym wykresie (rys.), posłużono się kalkulacją karty bankowej o nakładzie 100.000 sztuk, o wysokim stopniu zabezpieczenia (z jednoczesnym zastosowaniem opisanych powyżej metod, ale nie karty elektronicznej).

* * *

Systemy magnetycznych kart płatniczych, pomimo powszechnej, światowej akceptacji stoją wobec konieczności zasadniczych zmian w swojej strukturze, a wprowadzenie nawet uznanych produktów kartowych do Polski powinno wiązać się z weryfikacją stosowanych zabezpieczeń. Analizę skuteczności i zakresu oddziaływania poszczególnych metod zabezpieczeń zestawiono w tabeli 2.

Tabela 2. Analiza skuteczności metod zabezpieczeń kart, stosowanych w punktach używających EFTPOS

Zabezpieczenie przez:	Autentyczność karty	Tożsamość okaziciela	Prawdziwość akceptanta	Posiadane środki
Większa częstotliwość autoryzacji	+	-	-	+
Personalizacja zdjęcia na karcie	-	+	-	-
Sprawdzanie PIN-u przy transakcji	-	+ (*)	-	-
Wprowadzanie kodów MM, SECO, WATERMARK	+	-	-	-
Zastosowanie pomiarów biometrycznych	-	+	-	-
Dodanie układu chip	+	+(*)	+(&)	+ (^)

Źródło: opracowanie własne
Opis użytych oznaczeń:
+ weryfikuje
- nie weryfikuje
(*) weryfikuje tylko znajomość poprawnego PIN-u, nie zaś tożsamość okaziciela
(^) werfikuje (o ile przewidziano taką opcję w systemie)
(&) weryfikuje tylko terminal akceptanta

Scentralizowany system on-line wymaga wytworzenia infrastruktury międzybankowej niezbędnej do jego prawidłowego działania. Składają się na nią rozległe, szybkie i pojemne sieci transmisji oraz przetwarzania danych, zdolne do autoryzacji i realizacji płatności z każdego punktu przyjmującego transakcję, oraz niezbędne urządzenia do realizacji transakcji kartowych. Brak sprawnych łączy teletechnicznych może być poważnym problemem. Dodatkowo w warunkach naszego kraju trudno będzie dopasować wiele różnych systemów bankowych do operacji międzybankowych (np. dostępności danych o wszystkich kontach na poziomie oddziału każdego banku).

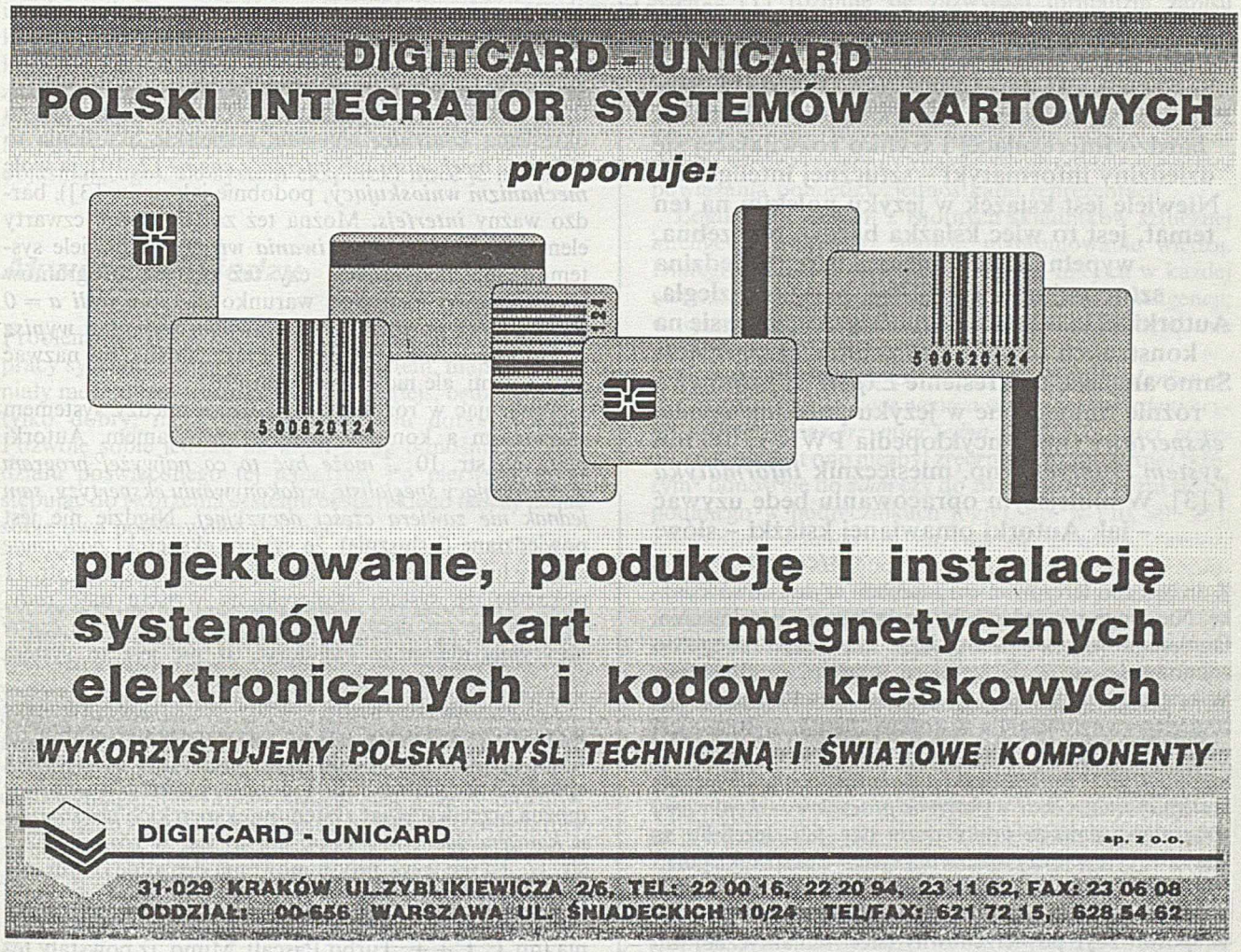
Rozproszony bezpieczny system off-line, nie wymagający w praktyce scentralizowanej autoryzacji kart jest praktycznie możliwy do zrealizowania jedynie przy wykorzystaniu systemu kart elektronicznych, w których pełna autoryzacja następuje w punkcie zawarcia transakcji, bez kontaktu z bankiem. Szacuje się, że wprowadzenie kart elektronicznych jako technologii następnej po kartach magnetycznych mogłoby zredukować do minimum nadużycia w okresie najbliższych 25 lat.

Wybór najodpowiedniejszego sposobu zabezpieczenia kart płatniczych w Polsce powinien się oprzeć na wnikliwej analizie możliwości zastosowania układu elektronicznego jako metody najnowocześniejszej oraz najbardziej przyszłościowej (w tej chwili). Jest to tym ważniejsze, że w świetle wspólnej publikacji VISA INT i MASTER CARD będzie to metoda proponowana i stosowana w przyszłości przez tych wydawców kart. Problem dobo-

ru właściwego zabezpieczenia powinien zostać rozwiązany przed dokonaniem pierwszych inwestycji w urządzeniach systemów kartowych, w przeciwnym razie będziemy zmuszeni za kilka lat do wymiany wszystkich urządzeń typu EFTPOS czy CD/ATM.

LITERATURA

[1] Andersson R.J.: UEPS – A second generation electronic wallet. Cambridge: University of Cambridge Computer Laboratory 1992
[2] Banking Technology No 2, pp. 30–32, 1993
[3] Binder P.: Chip Card – The competitive Edge, ESCAT conference proceedings. Helsinki 7–9.09.1994
[4] Card World User Guide 1993, International Journal for the Plastic Card, Financial & Retail Services Industries – Card World Publications Ltd., London, 1994
[5] Fraud Watch, Card World Publications Ltd. UK 1993
[6] Malicki A.: Raport z realizacji projektu – karta płatnicza. Lublin 1994
[7] Svigals J.: Magnetic Stripes, Twenty Fifth Anniversary. ESCAT Conference Proceedings. Helsinki, 7–9.09.1994
[8] Svigals J.: Smart cards – a security assessment Jerome Svigals Inc., 1992
[9] Townend R.: Electronic purse – The issues. Smart Card'94 Conference Proceedings, 15–17 February 1994. □



O/24/95

Deloitte & Touche Thomatsu International

Europejski kodeks dla superautostrady informatycznej

Komisja Europejska zaniepokojona brakiem regulacji prawnych w dziedzinie publikowania elektronicznego, zleciła firmie Deloitte & Touche Tohmatsu International przeanalizowanie kwestii związanych z uregulowaniami prawnymi, technologią, billingiem i marketingiem w odniesieniu do superautostrady informatycznej. Kierujący tym projektem David Hearn stwierdził, że Europę czeka jeszcze dużo pracy zanim w świecie biznesu będzie można w pełni wykorzystać potencjał tkwiący w tzw. korporacyjnym publikowaniu elektronicznym – ECP (*Electronic Corporate Publishing*). Firmy europejskie mogą uzyskać w sieci Internet informacje o ECP pod adresem **(http//www.internet-eireann.ie/dttirl** oraz zapisać się do grupy dyskusyjnej stworzonej w Internecie pod adresem **alt.publish.electronic.** Bliższych informacji na ten temat udziela też David Hearn: **e-mail David Hearn ₄dtti.team400.ie (k)**

ODA

Szesnaście współtworzących rynek firm z branży technologii informacyjnych, rozrywek, telekomunikacji, usług bankowych i handlowych poinformowało o powołaniu Porozumienia na Rzecz Definicji Obiektowych (*Object Definition Alliance* – ODA). Koalicja zamierza przyspieszyć rozwój multimedialnych usług interakcyjnych, ujednolicić standardy usług dokumentacyjnych i aplikacji wspomagających dokonanie transakcji przy pomocy mediów elektronicznych, rozbudzić zainteresowanie rynku tymi usługami oraz ustanowić otwarte i powszechnie dostępne specyfikacje obiektów aplikacji. Inicjatywa zawarcia porozumienia ODA wyszła od Oracle Corp.

Trudno jest pisać o sztucznej inteligencji (1)

System ekspercki, akwizycja i reprezentacja wiedzy, wnioskowanie

Halina Kwaśnicka
Centrum Informatyczne
Politechnika Wrocławska

Pojawiła się w księgarniach książka[1] dotycząca bardzo interesującej i szybko rozwijającej się dziedziny informatyki – sztucznej inteligencji. Niewiele jest książek w języku polskim na ten temat, jest to więc książka bardzo potrzebna, wypełniająca istniejącą lukę. Dziedzina sztucznej inteligencji jest bardzo rozległa, Autorki tej książki zapowiadają skupienie się na konstrukcji i analizie systemów *eksperckich*. Samo angielskie określenie *Expert System* bywa różnie tłumaczone w języku polskim: *system ekspertowy* (np. Encyklopedia PWN i [9]) lub *system ekspercki* (np. miesięcznik *Informatyka* i [3]. W niniejszym opracowaniu będę używać – jak Autorki omawianej książki – słów: *system ekspercki*.

Z dużym zainteresowaniem zaczęłam czytać nową książkę. Nie mogę powiedzieć, by spełniła moje oczekiwania. Ogólnie mówiąc, obawiam się, że nie zachęci nikogo do zajmowania się – fascynującą przecież – dziedziną. W książce, poświęconej systemom eksperckim, mówi się, że najlepsza sytuacja dla wdrożenia takiego systemu jest wtedy, gdy ma on ... *przejąć czynności, które pracownicy wykonują poza swoimi zwykłymi obowiązkami, lub też gdy system ma umożliwić wykonanie pracy, nie wykonywanej wcześniej z powodu braku czasu* (str. 52). Jeżeli to są właściwe sytuacje dla wdrożenia systemu eksperckiego, to trudno jest docenić ich prawdziwą rolę...

Niektóre działy w omawianej książce przedstawione są w sposób zbyt powierzchowny (sieci semantyczne) lub wręcz niepoprawny – np. niepewność w systemach eksperckich, gdzie, moim zdaniem, jest umieszczony błędny przykład. W dalszej części postaram się odnieść do pewnych problemów, budzących moje wątpliwości.

System ekspercki

Autorki często uciekają od przyjętego określenia system *ekspercki* pisząc *doradczy*. Wydaje mi się, że **system ekspercki jest co prawda systemem doradczym, ale niekoniecznie system doradczy musi być systemem eksperckim.** Zgodnie z definicją ze str. 9 książki, system ekspercki musi mieć trzy oddzielne elementy: **bazę wiedzy, mechanizm wnioskujący** (Autorki używają tu określenia ***kontroler wywodu***; angielskie określenia to *inference mechanism* lub *inference engine* [2, 10]; ja wolę ***mechanizm wnioskujący***, podobnie jak np. w [3]), bardzo ważny ***interfejs***. Można też za [5] dodać czwarty element – ***moduł pozyskiwania wiedzy***. Jest wiele systemów informatycznych, czy też małych programów (zawierających instrukcję warunkową typu: ***jeśli a = 0 wypisz „weź parasol" w przeciwnym wypadku wypisz „zostaw parasol w domu"***), które można by nazwać doradczymi, ale nie eksperckimi [10].

Analizując w rozdziale 2.1 różnice między systemem eksperckim a konwencjonalnym programem, Autorki piszą na str. 10 ... *może być to co najwyżej program wspomagający specjalistę w dokonywaniu ekspertyzy, sam jednak nie zawiera części decyzyjnej*. Nigdzie nie jest powiedziane (i słusznie), że system ekspercki *podejmuje decyzję* lub zawiera *część decyzyjną*. Rysunek 1 na str. 11 pokazuje, że system ekspercki na wyjściu daje *rady, wnioski*, nie zaś decyzje. Zatem brak w systemie *części decyzyjnej* nie może świadczyć, iż nie jest to system ekspercki.

Tak na marginesie, to systemy eksperckie, pracujące w czasie rzeczywistym, wykorzystywane w sterowaniu, na wyjściu dają konkretną **decyzję** a nie **opinię**; są to jednak systemy specyficzne [6]. Podobnie, moim zdaniem, nie można przeciwstawiać systemów eksperckich *programom w konwencjonalnym języku*, jak to jest w rozdziale 2.3, str. 12. To nie język programowania decyduje o tym, czy system jest ekspercki czy też nie. System ekspercki może być napisany w konwencjonalnym języku programowania (np. C, C++, Turbo Pascal). Mimo, iż powstały też dla systemów eksperckich języki specjalistyczne (**LISP, PROLOG**), to język programowania nie jest cechą wyróżniającą je spośród innych systemów.

Zaletą ***platformy***, czyli ***szkieletowego systemu eksperckiego***, nazywanego też ***powłoką*** [3], jest głównie to, że można dołączyć do niego wiedzę z dowolnej dziedziny i mieć gotowy system ekspercki. Wiedza musi być reprezentowana w ściśle określonej formie, którą zawarty w szkielecie mechanizm wnioskujący potrafi przetwarzać (str. 15). Nie powinny jednak być wymagane zmiany interfejsu, zmniejszałoby to bowiem użyteczność szkieletowych systemów eksperckich. Za pierwszy szkieletowy system ekspercki uważany jest **EMYCIN** (od Empty **MYCIN**, czyli *pusty* **MYCIN**), który powstał w Stanford University z systemu **MYCIN** po usunięciu z niego bazy wiedzy [13]. W Polsce jest rozwijany Szkieletowy System Ekspertowy **PC-Shell**; w firmie **AITECH** w Katowicach powstają jego kolejne wersje.

[1] J. Chromiec, E. Strzemieczna, *Sztuczna Inteligencja. Metody konstrukcji i analizy systemów eksperckich*, Akademicka Oficyna Wydawnicza PLJ, Warszawa, 1994, seria Informatyka.

Rola interfejsu

Na str. 12 Autorki piszą, że interfejs może być zredukowany do minimum, choć może to spowodować, że użytkownik ... *nie będzie chciał korzystać z systemu z powodu niesprzyjającego środowiska zewnętrznego programu* ... Z założenia, z systemem eksperckim nie pracuje informatyk (lekarz z systemami **MYCIN, PUFF,** geolog z **PROSPECTOR,** itp.). Interfejs, jak w każdym dobrym programie powinien być „przyjazny", ale w systemach eksperckich jest dodatkowy wymóg [5]: aby zwiększyć zaufanie użytkownika do systemu, musi on – na żądanie użytkownika – wyjaśnić sposób dochodzenia do konkluzji. Wymóg możliwości wyjaśniania sposobu wnioskowania jest stawiany systemom eksperckim w każdej, znanej mi pozycji literatury. Forsyth [5] pisze wręcz, że system z bazą wiedzy (ang. *knowledge-base system*) może nie mieć wszystkich czterech składników (w tym interfejsu objaśniającego), ale **system ekspercki mieć je musi.**

Akwizycja wiedzy

Problem *akwizycji wiedzy* jest kluczowy dla właściwej pracy systemu eksperckiego. Nasz system, mający wspaniały mechanizm wnioskowania i interfejs, będzie na tyle tylko dobry, na ile dostarczymy mu dobrej wiedzy. Pozwolę sobie jednak na kilka uwag odnośnie do rozdziału poświęconego tej tematyce. Po pierwsze, przystępując do tworzenia systemu eksperckiego należy zadać sobie dwa podstawowe pytania:

- Czy system taki jest potrzebny (co oznacza, czy znajdą się jego nabywcy);
- Czy wykonanie takiego systemu jest możliwe.

Jeśli twórca ma pomysł na system i chce go zrealizować w celach komercyjnych, powinien – aby odpowiedzieć na pierwsze pytanie – rozpoznać rynek oraz przeprowadzić analizę możliwości budowy systemu [7, 13]. Gdy natomiast pracuje na zamówienie, wtedy powinien przede wszystkim zbadać, czy wykonanie systemu, odpowiadającego wymaganiom zamawiającego jest w ogóle możliwe.

Zgadzam się, że dana dziedzina mocą samej swej natury może nie sprzyjać wdrażaniu w niej systemu eksperckiego, ale w rozważaniach dotyczących „realności wykorzystania" systemów eksperckich (str. 51) podanie jako przykładu takiej dziedziny **medycyny** – zaskoczyło mnie. Pierwszy znany i długo stosowany system ekspercki to przecież **MYCIN** – stworzony właśnie dla medycyny! Rozdział 25 w [13], zatytułowany *Katalog Systemów Eksperckich*, zawiera alfabetyczny spis dziedzin i działających w nich systemów eksperckich. Jak pisze Autor, katalog ten nie jest kompletny; książka wydana była w roku 1986 (nie dysponuję nowszymi danymi), jednak można sobie wyrobić pewien pogląd na ten temat. I oto w dziedzinie **medycyna** wymienionych jest wręcz najwięcej systemów – 53, dalej kolejno: **nauki wojskowe – 26, elektronika – 22, chemia – 14, systemy komputerowe – 11, prawo – 10, technika kosmiczna – 7, geologia – 7, rolnictwo – 3, meteorologia – 2.**

Autorki piszą, że pacjentowi trudno jest zaufać komputerowi, ale przecież to nie komputer, lecz lekarz stawia diagnozę, komputer tylko pomaga lekarzowi podjąć decyzję, nie zastępuje go[2]. System ekspercki nie podejmuje decyzji, ale daje radę. W jaki sposób z tej rady skorzysta użytkownik, nie jest to już sprawą systemu ani nawet jego projektanta. Użytkownik, aby wyrobić sobie opinię o danej przez system radzie, może zapytać system o sposób jej stworzenia i prześledzić ścieżkę wnioskowania.

Reprezentacja wiedzy

W systemach eksperckich problemem jest zarówno *co reprezentować* (jaką wiedzę), jak i *jak reprezentować*, aby wiedzą tą system mógł manipulować. Przez ***reprezentację wiedzy*** [1] rozumie się zazwyczaj formalizm zapisu informacji używanych przez system, wraz ze sposobami operowania na zbiorze tych informacji. Elementy wiedzy (reprezentowanej w systemie) są zazwyczaj powiązane między sobą, odzwierciedlając związki tematyczne, zakresy pojęć itp. Wszystkie powiązania między „jednostkami wiedzy" powinny znaleźć swoje odbicie w postaci powiązania pomiędzy „jednostkami reprezentacji".

Logika jest jednym z istotnych składników sztucznej inteligencji, ponieważ ułatwia manipulowanie wiedzą. Dlatego też logice poświęca się trochę miejsca w każdej niemalże książce, dotyczącej sztucznej inteligencji; w omawianej tutaj też, choć mnie ten rozdział nie satysfakcjonuje.

Na str. 17 podano określenie ***zmiennej zdaniowej***: *Proste zdania twierdzące nazywa się zmiennymi zdaniowymi, którym można przypisać jedną z dwóch wartości, prawdę lub fałsz.* Jest ono niezbyt zręcznie sformułowane („którym" odnosi się do *zdań* czy do *zmiennych?*) i różni się od podanego w podręcznikach logiki, np. w [10, str. 25]: *Zmiennymi zdaniowymi nazywa się zmienne, zamiast których, jeśli podstawimy zdania, otrzymamy całość będącą również zdaniem. A więc np. w wyrażeniu* ***jeżeli p, to q,*** *zmienne* ***p*** *i* ***q*** *są zmiennymi zdaniowymi*[3].

Bardzo ważne dla reprezentacji wiedzy są relacje (o angielskich nazwach ***instance*** i ***is-a***), pozwalające na reprezentację *hierarchii*, a co za tym idzie – *dziedziczenie* [12]. Można tłumaczyć ***instance*** **(x,Y)** jako **x** ***jest elementem zbioru*** **Y**, ***is-a*** **(X,Y)** jako **X** ***jest podzbiorem zbioru*** ***Y,*** co pociąga za sobą, że elementy zbioru **X** mają wszystkie cechy elementów zbioru **Y** oraz cechy specyficzne tylko dla siebie (dla wszystkich elementów zbioru **X**). Zatem, jeśli *Kora* to będzie imię papugi, *papugi* – zbiór wszystkich *ptaków* należących do gatunku papuga, to można powiedzieć, że: ***instance*** (*Kora, papugi*) oraz ***is_a*** (*papugi, ptaki*). Przykład 3 na str. 19 można zilustrować rysunkiem 1, aby widoczniejsze były zalety dziedziczenia.

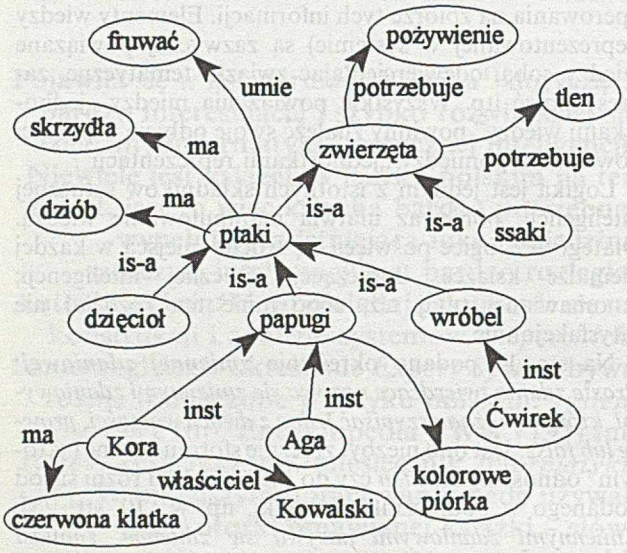
Z rysunku 1 widać, że zbiór *ptaki* jest podzbiorem zbioru *zwierzęta*, zatem elementy zbioru *ptaki* mają cechy *zwierząt*: taki element (czyli ptak) oddycha tlenem, potrzebuje pożywienia; zbiór *papugi* jest podzbiorem zbioru *ptaki*, każdy element zbioru *papugi* ma więc cechy *ptaków:*

---

[2] We Francji testowany był system ekspercki, pomagający podjąć decyzję lekarzowi w sali intensywnej terapii, dokąd czasami dowożony jest nieprzytomny pacjent (np. z ulicy). Bez możliwości przeprowadzenia wywiadu z pacjentem lekarz musi bardzo szybko podjąć decyzję o sposobie udzielania pomocy. System zbiera informacje z aparatury, do której podłącza się pacjenta oraz od samego lekarza i sugeruje podjęcie określonych działań. Opinie lekarzy testujących system były dość pozytywne (informacja pochodzi z filmu, pokazywanego bodajże w programie telewizyjnym KWANT, w ubiegłym roku).

[3] Mówiąc nawiasem, „uroda" tego zdania jest też mocno wątpliwa ... *(red.).*

ma skrzydła, dziób, znosi jaja, itp. *Kora* jest elementem zbioru *papugi*, ma zatem cechy wszystkich papug, ale ma też cechy indywidualne, odróżniające ją od innych papug, np. ma właściciela – Kowalskiego, czerwoną klatkę, itp. Chcąc zawrzeć w systemie atrybuty *Kory*, należy przy jej opisie umieścić tylko wiedzę dla niej charakterystyczną, odróżniającą ją od innych papug. Pozostałe cechy *Kory* będą wynikały z dziedziczenia: z tego, że jest ona papugą oraz z tego, że jest ona ptakiem, zwierzęciem, itd. Wiedza o atrybutach ptaków może być pamiętana w jednym miejscu, a nie przy każdym gatunku ptaków (papuga, dzięcioł, wróbel, itd.), co pozwala oszczędzać pamięć.

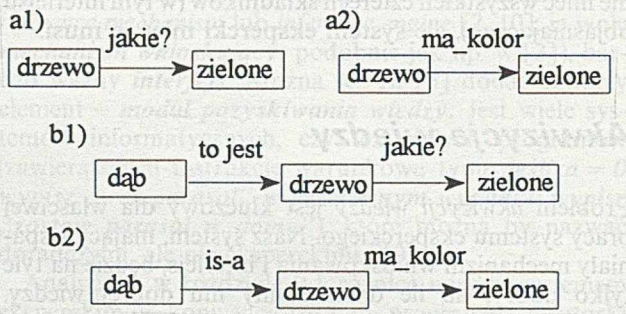


Rys. 1. Relacje *is-a* i *instance* w reprezentacji dziedziczenia

Sieci semantyczne (rozdział 5) są wygodną graficzną reprezentacją relacji (włącznie z własnością, przy czym własność rozumiana jest jako relacja jednoargumentowa) pomiędzy obiektami. Jak zauważyły Autorki, jest to technika rzadko używana, ponieważ dla rzeczywistych problemów sieć semantyczna musiałaby być bardzo duża i zarówno wykonanie takiej sieci, jak i przetwarzanie jej nastręcza trudności. Dyskusyjne jest jednak stwierdzenie Autorek, że sieci semantyczne ... *są dość ubogim narzędziem* (str. 39). Są one – przeciwnie – bardzo elastyczne i ułatwiają ujęcie całej wiedzy, potrzebnej do rozwiązywania określonych problemów. Sama definicja sieci semantycznej podana na str. 39 jest zagmatwana, trudno dopatrzyć się w niej, że łuki reprezentują relacje występujące pomiędzy obiektami. W definicji jest: ... *u: S → L funkcja określająca nazwę (własność) odpowiadającą połączeniu między węzłami (L – zbiór własności reprezentowanych przez sieć)*. Nie wiadomo, co *nazwa* reprezentuje. Rysunek 1 ze str. 40 też nic tu nie wyjaśnia, a nawet może wprowadzić w błąd.

Sieć semantyczna jest digrafem etykietowanym. Zarówno węzły, jak i łuki są etykietowane wybranymi nazwami. Węzły reprezentują zwykle obiekty fizyczne (zwierzęta, przedmioty) lub też pojęcia abstrakcyjne – akcje, sytuacje, pojęcia koncepcyjne, atrybuty (cechy charakterystyczne obiektów). Łuki najczęściej łączą węzły, reprezentujące obiekty, z węzłami, reprezentującymi atrybuty; reprezentują one różnego rodzaju relacje, w tym poznane już relacje ***instance*** (***jest_elementem***), ***is-a*** (***jest_podzbiorem*** – reprezentuje dziedziczenie) oraz inne, np. ***ma, jest_częścią, używa***, itp.

Wróćmy do rysunku 1, str. 40: w części a) są dwa węzły o nazwach ***drzewo*** i ***zielone***, połączone łukiem skierowanym (do węzła zielone) z napisem ***jakie?*** Spróbujmy dojrzeć tu jakąś relację – jest to dość trudne. Z czym (jakim innym węzłem) ***drzewo*** może być w relacji ***jakie?***, co to w ogóle za relacja ***jakie?*** Proponowałabym relację o nazwie ***ma_kolor*** (lub też ***jest_koloru***). W części b) mogę się domyślać, że chodzi nie o to, że ***dąb jest drzewem*** jak mówi podpis pod rysunkiem, ale o to, że ***dęby*** są podzbiorem ***drzew*** (zob. rys. 2); wtedy sieć będzie reprezentować hierarchię (każdy ***dąb*** ma cechy zbioru ***drzewa***). Relacja od węzła ***dąb*** do węzła ***drzewo*** będzie ***is-a***, natomiast od węzła drzewo do węzła zielone będzie ***ma_kolor***.



Rys. 2. Przykładowa sieć semantyczna; a1) i b1) – rys. 1, s. 40, a2) i b2) – propozycje nazw relacji

Na wszystkich sześciu rysunkach (str. 40–44) należałoby przyjąć jednakowe opisy: etykiety dla łuków pisać albo na rysunku (jak na rys. 1), albo w postaci legendy jak na rysunkach 3, 4 i 5, przy czym legenda do rysunków 4 i 5 nie jest pełna: brak opisu zaciemnionych prostokątów. Na rysunkach 2 i 6 opis relacji jest w treści podpisu pod rysunkami. Znaczenia zaciemnionych prostokątów wynikają z opisu przykładu, niemniej utrudnia to czytanie książki.

Wnioskowanie

W rozdziale 3.3. opisano ***metodę rezolucji*** jako efektywną metodę dowodzenia twierdzeń (dowodzenie przez zaprzeczenie). W następnym rozdziale (3.4) jest mowa o języku programowania w logice – **Prologu**, który ... *wydobywa za pomocą dedukcji logicznej również fakty nie podane explicite przez użytkownika* ... (str. 21). Ale co to jest *dedukcja?* Trudno zrozumieć dlaczego wyjaśnione jest pojęcie *rezolucji*, które dalej w pracy nie jest wykorzystywane, natomiast o wykorzystywanym nieco dalej pojęciu *dedukcja* nie ma ani słowa ...

Zgodnie z literaturą, *dedukcja* jest *uzasadnioną logicznie i systematyczną procedurą wnioskowania*, a *wnioskowanie* jest to *proces dedukcji nowych faktów z innych*, znanych faktów [3, str. 39]. Według [2, str. 14], *dedukcja* jest *wnioskowaniem logicznie poprawnym – „logically correct inference"*. Cały proces wnioskowania, stosowany nie tylko w **Prologu**, byłby bardziej zrozumiały dla Czytelników, gdyby w pracy znalazło się choć parę zdań na temat dedukcji i implikacji.

W procesie dedukcji wykorzystuje się implikację:

- zakłada się, że prawdziwa jest implikacja $p \Rightarrow q$ (przyjmujemy, że reguły wnioskowania, sformułowane w postaci IF *przesłanka* THEN *konkluzja*, zawarte w bazie wiedzy, są prawdziwe),
- sprawdza się, czy przesłanka p jest prawdziwa (sprawdzamy, czy prawdziwe są zdania występujące w części IF reguły, tzn. czy w bazie faktów systemu są odpowiednie fakty, lub przez zapytanie systemu, kierowane do użytkownika),
- jeśli tak, to konkluzja q musi być też prawdziwa (tzn. mamy wydedukowany nowy fakt, który jako prawdziwy – o wartości logicznej TRUE – dodawany jest do zbioru faktów).

Powyższe wynika z własności implikacji, która przy prawdziwej przesłance jest prawdziwa tylko wtedy, gdy konkluzja jest prawdziwa. Przesłanka p może być zdaniem złożonym, np. z dwóch zdań połączonych operatorem logicznym AND.

Wydedukowana w jednej regule konkluzja może być przesłanką w innej regule, można zatem z niej wydedukować kolejną konkluzję, która również może być przesłanką w innej regule, itd., aż nie będzie już reguł, które można będzie zastosować, lub też nastąpi inny, ustalony koniec wnioskowania. W ten sposób zachodzi proces wnioskowania „w przód", (ang. *forward chaining*). Do testowania hipotez (znamy cel, chcemy sprawdzić, czy jest on w świetle danej bazy wiedzy i faktów prawdziwy) stosuje się też wnioskowanie „w tył" (ang. *backward chaining*): szukamy reguł, które w części THEN mają jako konkluzję nasz cel i sprawdzamy, czy te reguły mogą być zastosowane (używa się też pojęcia *zapalone*). Mogą być zastosowane wtedy, gdy albo w bazie faktów są prawdziwe ich przesłanki, albo też mogą być one wydedukowane z innych reguł (są w części THEN innych reguł). Proces wnioskowania trwa, aż hipoteza zostanie potwierdzona lub zaprzeczona. O wnioskowaniu wstecz i w przód jest krótko powiedziane w książce tylko przy okazji omawiania narzędzia do tworzenia systemów eksperckich – w rozdziale 10.

Dalsza część omówienia książki ukaże się w następnym numerze Informatyki.

LITERATURA

[1] Bolc L., Cichy M., Różańska L.: Przetwarzanie języka naturalnego, WNT, Warszawa, 1982
[2] Charniak E., McDermott D.: Introduction to Artificial Intelligence, Addison-Wesley Pub. Company, 1985
[3] Chwałkowska E.: Sztuczna inteligencja w systemach eksperckich, MIKOM, Warszawa, 1991
[4] Clarke N. D., McLeish M. D., Vyn T. J., Stone J. A.: Using Certainty Factors and Possibility Theory Methods in a Tillage Selection Expert System; w: Expert Systems with Applications. An International Journal, Vol. 4, No 1, pp. 53–62, 1992
[5] Forsyth R.: Expert Systems. Principles and case studies, Chapman and Hall Ltd., New York, London, 1984
[6] Grzegorczyk A.: Zarys logiki matematycznej, PWN, 1973
[7] Kwaśnicka H., Markowska-Kaczmar U.: Sztuczna inteligencja w sterowaniu, Informatyka nr 6, s. 3–9, 1994
[8] Kowalski R.: Logika w rozwiązywaniu zadań, WNT, Warszawa, 1989
[9] Lech M. (red.): Elementy Systemów Ekspertowych. Skrypty AE, Wrocław, 1991
[10] Levine R. I., Drang D. E., Edelson B.: A Comprehensive Guide to AI and EXPERT SYSTEMS: Turbo Pascal Edition, McGraw-Hill Book Company, USA, 1988
[11] Rich E., Knight K.: Artificial Intelligence, second ed., McGraw-Hill. Inc., 1989
[12] Schalkoff R. J.: Artificial Intelligence: An Engineering Approach, McGraw-Hill Book Company, USA, 1990
[13] Waterman D. A.: A Guide to Expert Systems, Addison-Wesley Pub. Company, 1986

0/26/95

Quarterdeck proponuje:
Usprawnienia Windows

Amerykańska firma Quarterdeck, której dystrybutorem w Polsce jest warszawska firma SoftPoint, wprowadziła do sprzedaży kilka produktów, które ułatwiają i przyspieszają pracę z Windows, a szczególnie z Windows 95. CleanSweep 95 to prosty w obsłudze i w pełni zautomatyzowany deinstalator do każdej wersji Windows, który usuwa szybko i bezpiecznie pliki 16- i 32-bitowych aplikacji. MagnaRAM 2 – *memory multipler* – udostępnia programom Windows maksimum dostępnej pamięci (potrafi ją trzykrotnie zwiększyć). WinProbe 4 – to automatyczny optymalizator zasobów systemu i pamięci. Zawiera m.in. pełny pakiet funkcji diagnostycznych, automatycznie optymalizuje konfigurację systemu, zapewnia ponad 300 porad i wskazówek, jak usprawnić system i ponad 150 wskazówek specjalnie dla Windows 95. InternetSuite – to kompleksowy pakiet do pracy z Internetem, obsługujący połączenia przez modem. (k)

POLIXEL

SYSTEMY PREZENTACYJNE LCD

InFocus SYSTEMS
⇒ *Projektory multimedialne*
⇒ *Panele ciekłokrystaliczne*

KOMUNIKACJA LAN / WAN

RAD
⇒ *Modemy pasma podstawowego*
⇒ *Sieci pakietowe*
⇒ *Multipleksery dane/głos*
⇒ *Zdalny dostęp do sieci LAN*

LANNET
⇒ *Huby i Switche*

RND RAD Network Devices
⇒ *Mosty i Routery*

RADCOM
⇒ *Analizatory WAN/LAN/ATM*

RAD LinX
⇒ *Serwery terminali i drukarek*

RiT
⇒ *Krosownice zarządzane SNMP, osprzęt sieciowy*

Silicom
⇒ *Adaptery sieciowe PCMCIA*

Zapewniamy Wsparcie Techniczne Na Którym Możesz Polegać

POLIXEL S. A. Ul. Zakopiańska 6, PL 03-934 Warszawa,
Tel. (48-22) 617 83 81, Tel./Fax (48-22) 617 90 01

Węzły wirtualne
jako metoda zwiększenia odporności na błędy aplikacji działających w systemach rozproszonych

Henryk Krawczyk
Piotr W. Umiński
Katedra Architektury Systemów Komputerowych
Politechnika Gdańska

Obserwujemy obecnie ciągły wzrost znaczenia systemów rozproszonych. Sieciowe systemy rozproszone pozwalają zarówno na zwiększenie efektywności przetwarzania za cenę niższą niż istniejące superkomputery, jak i na znacznie lepsze wykorzystanie posiadanych zasobów. Co więcej, zastosowanie systemów rozproszonych staje się niezbędne do zapewnienia dostępu do różnego rodzaju informacji z wielu punktów jednocześnie, na przykład w bankowości czy transporcie lotniczym i morskim. W tego typu aplikacjach wymagania niezawodnościowe odgrywają coraz istotniejszą rolę. W pracy zaproponowano nową koncepcję tworzenia systemów rozproszonych, zapewniających zwiększoną odporność na błędy sprzętowe i programowe przy niezbyt wysokich kosztach budowy systemu. Omawiane podejście może być zastosowane przede wszystkim w dużych systemach sieciowych wykonujących równocześnie zadania o różnym charakterze.

Prace nad systemami o zwiększonej odporności na błędy są dość zaawansowane. Osiągnięcia w tej dziedzinie, dotyczące systemów jednoprocesorowych oraz dedykowanych systemów wieloprocesorowych są znaczne. Dużo trudniejsze jest zapewnienie odporności na błędy systemów rozproszonych. Istnieje wiele strategii diagnozowania systemów rozproszonych, podobnie jak wiele strategii odtwarzania poprawnego stanu systemu po awarii węzła. Niestety, optymalne rozwiązanie problemu zapewnienia odporności na błędy nie polega na prostym połączeniu optymalnej strategii diagnozowania systemu i optymalnej strategii odtwarzania stanu po awarii [3]. Powodem tego jest fakt, że architektura systemu (a więc struktura połączeń, nadmiarowość sprzętowa itp.) dająca największe możliwości diagnozowania nie musi być optymalna ze względu na odtwarzanie poprawnego stanu. Poza tym łączny koszt diagnozowania i odtwarzania w systemie może się okazać zbyt wysoki i przez to niemożliwy do zaakceptowania.

Próby zastosowania zintegrowanej strategii diagnozowania i odtwarzania omówione zostały w pracach [2] i [3]. Polegają one – najogólniej – na testowaniu poprawności systemu przez wykonywanie tego samego zadania jednocześnie na kilku różnych węzłach. Porównanie wyników otrzymanych przez poszczególne węzły daje możliwość wyodrębnienia węzłów uszkodzonych. Pozwala to jednocześnie na narzucenie wyników zaakceptowanych jako poprawne przez większość stacji wszystkim pozostałym węzłom.

Pomimo licznych zalet, rozwiązania te mają – jak się wydaje – poważne ograniczenie. Zakładają bowiem, że w systemie można wydzielić odrębne zadania, które mogą być wykonywane niezależnie w kilku wersjach na oddzielnych węzłach, bez konieczności współpracy z innymi zadaniami. Zadania te produkują określone zbiory wyników, które możemy porównywać. Niestety, w rzeczywistych systemach taka dekompozycja jest przeważnie dość trudna i nieefektywna. W zrealizowanych systemach komercyjnych częściej mamy do czynienia z kilkoma procesami, które osadzone na różnych węzłach działają przez dłuższy czas, i które w tym czasie komunikują się między sobą. Można wprawdzie z każdego z tych procesów wyodrębnić „zadania", trwające od jednej komunikacji międzyprocesowej do drugiej, ale rozwiązanie takie jest nienaturalne i nieczytelne. Narzucanie ze względu na bezpieczeństwo systemu struktury nieodpowiedniej dla danego problemu może prowadzić wręcz do pogorszenia jego bezpieczeństwa, przez zwiększenie prawdopodobieństwa powstania błędów w fazie projektowania oraz realizacji.

Alternatywnym rozwiązaniem wydawać by się mogło zwielokrotnianie elementów istniejących systemów. Nadmiarowe rozwiązania sprzętowe (w postaci modularności funkcjonalnej) i programowe (w postaci bloków odtworzeniowych) są stosowane w systemach scentralizowanych [1]. Drogą powielania sprzętu można również uzyskać większą niezawodność pojedynczego węzła systemu rozproszonego. Zastosowanie zamiast jednego – kilku komputerów, wykonujących jednocześnie to samo zadanie, albo napisanie kilku różnych procedur, wykonujących tę samą operację, pozwala na znaczne zwiększenie niezawodności działania, bez konieczności zmiany koncepcji systemu w skali makro. Niestety, koszty takiego podejścia są z reguły wysokie. Konieczność zakupu „na wszelki wypadek" dwu- trzykrotnie większej liczby sprzętu zniechęci do projektu nawet najbardziej rozrzutnego i przezornego menedżera. Podobnie rzecz się ma z kosztami wytwarzania dodatkowego oprogramowania.

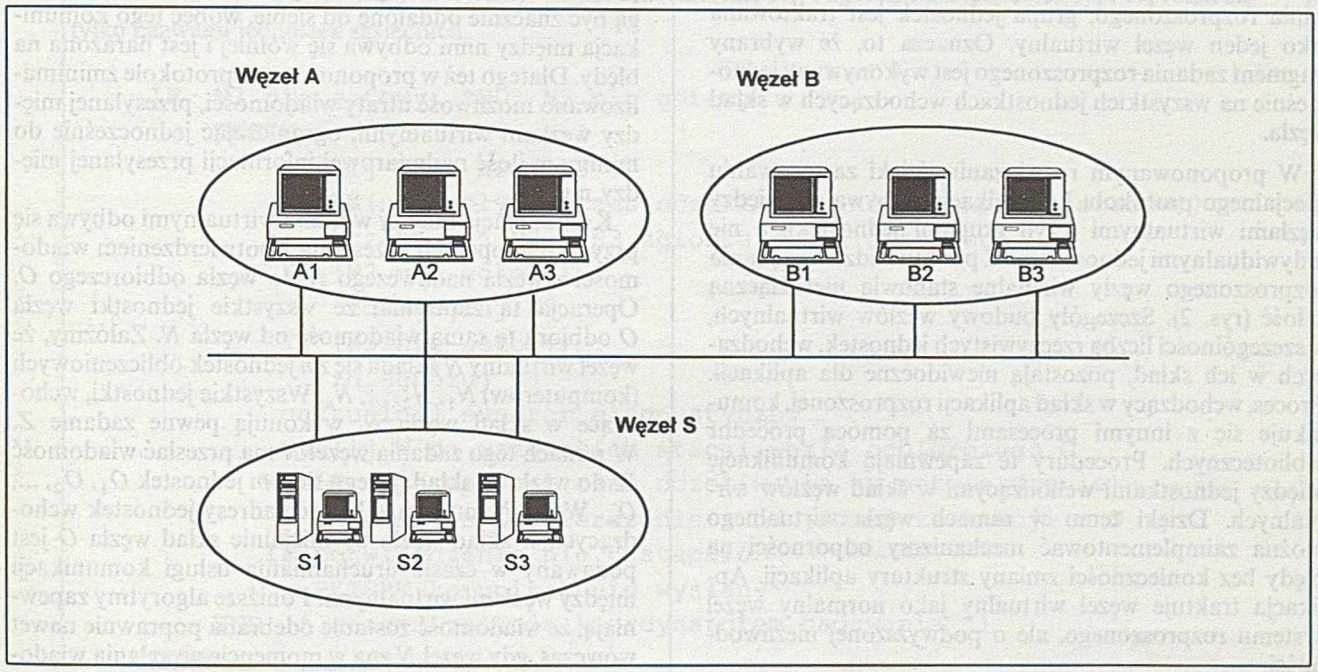
W tej sytuacji należy szukać rozwiązań pośrednich: ograniczających kosztowną nadmiarowość w systemie do rzeczywiście niezbędnych rozmiarów, a jednocześnie dających projektantom oprogramowania użytkowego możliwość łatwego wykorzystania koncepcji już istniejących. Oryginalną metodę odtwarzania wstecznego w systemie rozproszonym zaprezentowano w pracy [6]. Praca niniejsza przedstawia natomiast koncepcję systemu informaty-

cznego, w którym zwiększoną odporność na błędy uzyskuje się na drodze odtwarzania po awarii bieżącego poprawnego stanu. Zaproponowana metoda pozwala na racjonalne zagospodarowanie zasobów już istniejących systemów komputerowych.

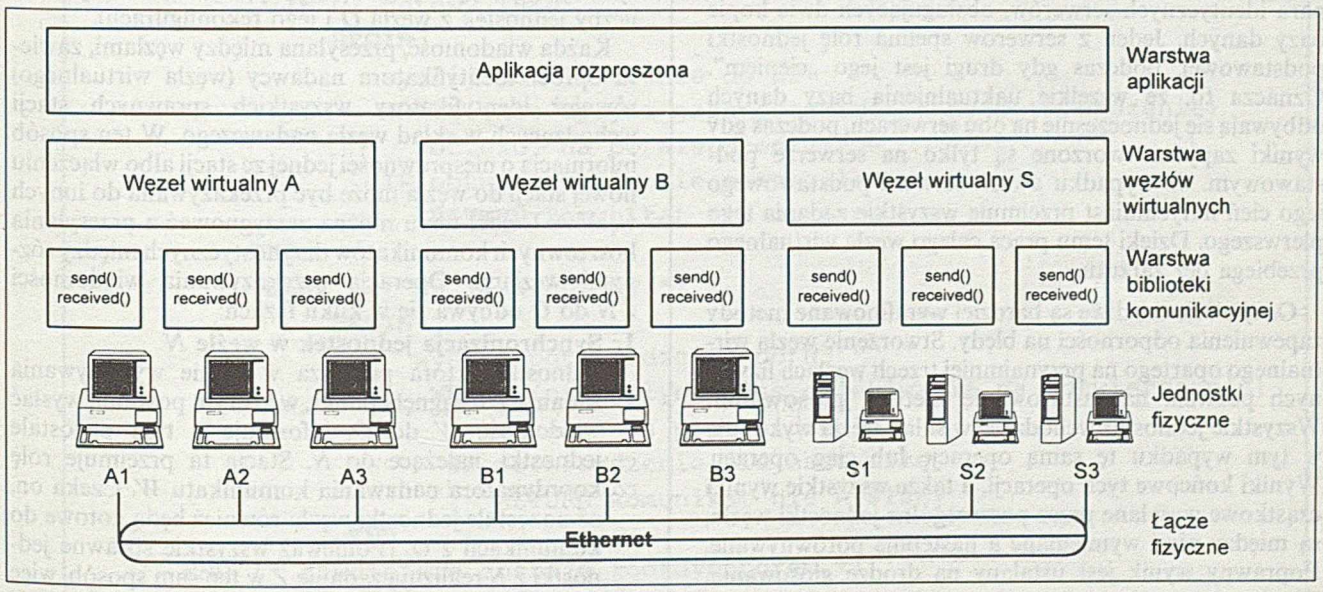
Koncepcja węzłów wirtualnych i ich struktura

W systemie rozproszonym mamy do czynienia z wieloma komputerami (węzłami), często odległymi od siebie, które mogą się komunikować za pośrednictwem linii transmisyjnych. W dalszym ciągu naszych rozważań będziemy zajmować się tylko systemami komputerowymi, w których poszczególne jednostki (komputery) działają pod kontrolą wielozadaniowego systemu operacyjnego (np. systemu UNIX) i są połączone siecią komputerową. Zastosowany protokół komunikacyjny (np. TCP/IP) pozwala na zestawianie wirtualnych połączeń między dowolnymi węzłami sieci. Systemy takie, w których węzłami są uniwersalne stacje robocze, są najczęściej stosowane jako rozwiązania najtańsze i najbardziej elastyczne.

W omawianym środowisku mamy do czynienia z dwoma rodzajami zadań: zadaniami wykonywanymi tylko przez jeden węzeł sieci (nazywanych dalej **zadaniami lokalnymi**) oraz z zadaniami wymagającymi współpracy wielu węzłów (nazywanych **zadaniami rozproszonymi**). Do pierwszej grupy zadań zaliczyć można edycję tekstów, prace projektowe przy użyciu narzędzi typu CAD/CAM, czy prace wykonywane przez studentów w sieci akademickiej. Z kolei zadania angażujące wiele węzłów to, na przykład, operacje przeszukiwania i uaktualniania rozproszonej bazy danych, czy wykonywanie skomplikowanych obliczeń naukowych, wykorzystujących algorytmy równoległe. W większości wypadków w omawianych



Rys. 1. Struktura fizyczna systemu komputerowego z wyodrębnionymi węzłami wirtualnymi



Rys. 2. Struktura systemu rozproszonego opartego na węzłach wirtualnych

systemach komputerowych wszystkie te zadania są wykonywane równocześnie. Należy zwrócić uwagę, że poszczególne zadania mają różne wymagania odnośnie do niezawodności systemu. O ile upadek stacji roboczej w czasie edycji tekstu powoduje co najwyżej utratę fragmentu tekstu i opóźnienie w pracy jednej osoby, to upadek tej stacji podczas wykonywania skomplikowanej transakcji w bankowej bazie danych może spowodować wielogodzinny przestój całego systemu i olbrzymie straty finansowe.

Koncepcja węzłów wirtualnych polega na logicznym wyodrębnieniu z systemu komputerowego grup kilku (co najmniej dwu) jednostek o podobnych parametrach (patrz rys. 1). W celu zwiększenia efektywności węzły te powinny być połączone za pomocą szybkiego złącza, np. siecią lokalną. Każda jednostka, wchodząca w skład takiej grupy, może wykonywać swoje zadania lokalne. Jednocześnie, w przypadku wykonywania fragmentu zadania rozproszonego, grupa jednostek jest traktowana jako jeden węzeł wirtualny. Oznacza to, że wybrany fragment zadania rozproszonego jest wykonywany jednocześnie na wszystkich jednostkach wchodzących w skład węzła.

W proponowanym rozwiązaniu dzięki zastosowaniu specjalnego protokołu komunikacja odbywa się między węzłami wirtualnymi (czyli grupami jednostek), a nie indywidualnymi jednostkami. Z punktu widzenia zadania rozproszonego węzły wirtualne stanowią nierozłączną całość (rys. 2). Szczegóły budowy węzłów wirtualnych, w szczególności liczba rzeczywistych jednostek, wchodzących w ich skład, pozostają niewidoczne dla aplikacji. Proces, wchodzący w skład aplikacji rozproszonej, komunikuje się z innymi procesami za pomocą procedur bibliotecznych. Procedury te zapewniają komunikację między jednostkami wchodzącymi w skład węzłów wirtualnych. Dzięki temu w ramach węzła wirtualnego można zaimplementować mechanizmy odporności na błędy bez konieczności zmiany struktury aplikacji. Aplikacja traktuje węzeł wirtualny jako normalny węzeł systemu rozproszonego, ale o podwyższonej niezawodności.

Najprostszym przykładem węzła wirtualnego może być para identycznych serwerów, obsługujących dwie kopie bazy danych. Jeden z serwerów spełnia rolę jednostki podstawowej, podczas gdy drugi jest jego „cieniem". Oznacza to, że wszelkie uaktualnienia bazy danych odbywają się jednocześnie na obu serwerach, podczas gdy wyniki zapytań tworzone są tylko na serwerze podstawowym. W wypadku awarii serwera podstawowego jego cień natychmiast przejmuje wszystkie zadania tego pierwszego. Dzięki temu praca całego węzła wirtualnego przebiega bez zarzutu.

Oczywiście możliwe są bardziej wyrafinowane metody zapewnienia odporności na błędy. Stworzenie węzła wirtualnego opartego na przynajmniej trzech węzłach fizycznych pozwala na zastosowanie **metody głosowania.** Wszystkie jednostki wchodzące w skład węzła wykonują w tym wypadku tę samą operację lub ciąg operacji. Wyniki końcowe tych operacji, a także wszystkie wyniki cząstkowe wysyłane przez poszczególne jednostki węzła, są między nimi wymieniane a następnie porównywane. Poprawny wynik jest ustalany na drodze głosowania. Wynik ten jest następnie przekazywany na zewnątrz jako poprawny wynik całego węzła wirtualnego.

Protokół komunikacji między węzłami wirtualnymi

Podstawową operacją, która musi być dostępna w systemie z węzłami wirtualnymi, jest operacja przesłania wiadomości między dwoma węzłami wirtualnymi. Na jej podstawie mogą być realizowane bardziej rozbudowane protokoły komunikacji między węzłami.

Omawiany protokół zakłada połączenie wszystkich stacji siecią komputerową. W odróżnieniu jednak od proponowanej w pracy [5] koncepcji komunikacji, wzorowanej na grupach dyskusyjnych w sieci Internet, omawiany protokół nie stosuje rozgłaszania, a tylko wymianę komunikatów między poszczególnymi stacjami.

Zakładamy, że jednostki wchodzące w skład węzła wirtualnego są połączone siecią lokalną, dzięki czemu komunikacja między nimi odbywa się szybko i bez przekłamań. Ponieważ poszczególne węzły wirtualne mogą być znacznie oddalone od siebie, wobec tego komunikacja między nimi odbywa się wolniej i jest narażona na błędy. Dlatego też w proponowanym protokole zminimalizowano możliwość utraty wiadomości, przesyłanej między węzłami wirtualnymi, ograniczając jednocześnie do minimum ilość nadmiarowej informacji przesyłanej między nimi.

Komunikacja między węzłami wirtualnymi odbywa się przy użyciu operacji przesłania z potwierdzeniem wiadomości z węzła nadawczego N do węzła odbiorczego O. Operacja ta zapewnia, że wszystkie jednostki węzła O odbiorą tę samą wiadomość od węzła N. Załóżmy, że węzeł wirtualny N składa się z n jednostek obliczeniowych (komputerów) $N_1, N_2, ..., N_n$. Wszystkie jednostki, wchodzące w skład węzła N, wykonują pewne zadanie Z. W ramach tego zadania węzeł N ma przesłać wiadomość W do węzła O, składającego się z m jednostek $O_1, O_2, ..., O_m$. Węzeł N musi znać bieżące adresy jednostek wchodzących w skład węzła O. Inicjalnie skład węzła O jest podawany w czasie uruchamiania usługi komunikacji między węzłami wirtualnymi. Poniższe algorytmy zapewniają, że wiadomość zostanie odebrana poprawnie nawet wówczas, gdy węzeł N zna w momencie wysyłania wiadomości tylko niektóre z bieżących adresów węzła O (taka sytuacja może powstać np. w wypadku awarii pewnej liczby jednostek z węzła O i jego rekonfiguracji).

Każda wiadomość, przesyłana między węzłami, zawiera oprócz identyfikatora nadawcy (węzła wirtualnego) również identyfikatory wszystkich sprawnych stacji wchodzących w skład węzła nadawczego. W ten sposób informacja o niesprawności jednej ze stacji albo włączeniu nowej stacji do węzła może być przekazywana do innych węzłów. Dzięki temu można zrezygnować z przesyłania kosztownych komunikatów diagnostycznych między różnymi węzłami. Operacja przekazywania wiadomości z N do O odbywa się w kilku fazach:

1. **Synchronizacja jednostek w węźle N**
   Jednostka, która pierwsza w czasie wykonywania zadania Z osiągnęła punkt, w którym powinna wysłać wiadomość W do O, informuje o tym pozostałe jednostki, należące do N. Stacja ta przejmuje rolę koordynatora nadawania komunikatu W. Czeka on, aż pozostałe jednostki węzła również będą gotowe do komunikacji z O. (Ponieważ wszystkie sprawne jednostki z N realizują zadanie Z w ten sam sposób, więc taka synchronizacja na pewno nastąpi). W momencie stwierdzenia gotowości wszystkich jednostek następu-

je uzgodnienie treści komunikatu *W* wysyłanego do *O*. Odbywa się to na drodze głosowania rozproszonego. Dzięki temu jest określany również stan sprawności poszczególnych jednostek i na tej podstawie jest możliwa rekonfiguracja węzła.

2. **Przesłanie wiadomości do *O***
   Jednostka – koordynator przesyła wiadomość *W* do węzła *O*. Odbywa się to przez wysłanie wiadomości *W* do wszystkich jednostek z *O*. Razem z informacją użytkową jest przesyłany również status węzła *N* w postaci adresów wszystkich sprawnych jednostek wchodzących w tym momencie w skład *N*.
3. **Rozesłanie wiadomości w węźle *O***
   Jednostki w węźle *O* przekazują sobie komunikat *W*. Jednostka, która odebrała wiadomość, retransmituje ją do pozostałych jednostek. Dzięki temu nawet w wypadku, gdy pewna jednostka z *O* nie otrzyma wiadomości bezpośrednio od nadawcy, otrzyma ją od sąsiednich jednostek swojego węzła. Ta faza może być również wykorzystywana do diagnozowania jednostek wchodzących w skład *O*.
4. **Potwierdzenie przyjęcia wiadomości przez *O***
   Jednostka, która pierwsza odebrała wiadomość od węzła *N*, zostaje koordynatorem odbioru. Jej zada-

ALGORYTM 1:
Synchronizacja jednostek nadawczych podczas wysyłania wiadomości W z węzła wirtualnego N do węzła wirtualnego O

UWAGA: Poniższy algorytm jest wykonywany przez jednostkę N_1. Algorytm dla jednostek N_2 i N_3 różni się tylko nazwami jednostek sąsiednich.

```
IF  N2 nie gotowy  AND  N3 nie gotowy
    BEGIN
        wyślij W do N2 i N3
        WHILE(nie upłynął czas oczekiwania na gotowość sąsiadów))
            czekaj aż N2 i N3 zgłoszą swoją gotowość
        IF (N2 nie gotowy)
            zdiagnozuj N2
        IF (N3 nie gotowy)
           zdiagnozuj N3
        uaktualnij syndrom diagnostyczny
        wyślij W do wszystkich stacji węzła odbiorczego O
       WHILE (nie upłynął czas oczekiwania na potwierdzenie)
          oczekuj na potwierdzenie z od jednej ze stacji z O
       IF potwierdzenie nie nastąpiło
          RETURN komunikat nie wysłany
    END (* stacja zostaje koordynatorem nadawania *)
ELSE
    BEGIN
        IF N2 gotowy THEN
           BEGIN
              wyślij potwierdzenie do N2
              WHILE (nie upłyną czas oczekiwania na potwierdzenie)
                 oczekuj na potwierdzenie z O
              IF potwierdzenie nie nastąpiło
                 RETURN komunikat nie wysłany
      END
       ELSE  (* N3 jest gotowy *)
           BEGIN
                wyślij potwierdzenie do N3
                WHILE (nie upłyną czas oczekiwania na potwierdzenie)
                    oczekuj na potwierdzenie z O
                 IF potwierdzenie nie nastąpiło
                    RETURN komunikat nie wysłany
            END
  END (* stacja nie zostaje koordynatorem *)
```

ALGORYTM 2:
Algorytm synchronizacji jednostek węzła O przy odbiorze wiadomości z węzła N

UWAGA: Poniższy algorytm jest wykonywany przez jednostkę O_1. Algorytm dla jednostek O_2 i O_3 różni się tylko nazwami jednostek sąsiednich.

```
ustaw maksymalny czas synchronizacji odbioru na Tmax
WHILE (synchronizacja  nie zakończona)
  odczytaj kolejną wiadomość
  CASE typ otrzymanej wiadomości OF:
    wiadomość W od nadawcy N:
        BEGIN
          zapamiętaj wiadomość W
          IF O1 jest pierwszym węzłem który otrzymał W THEN
           O1 zostaje koordynatorem odbioru
          IF O2 nie potwierdził odebrania wiadomości THEN
             retransmituj W do O2
          IF O3 nie potwierdził odebrania wiadomości THEN
             retransmituj W do O3
        END;
    wiadomość W retransmitowana przez odbiornik O2:
        BEGIN
          IF W jeszcze nie zapamiętana THEN
            zapamiętaj W
          wyślij potwierdzenie do O2
          IF O3 się jeszcze nie odezwał THEN
            retransmituj W do O3
        END;
    wiadomość W retransmitowana przez odbiornik O3
        BEGIN
         IF W jeszcze nie zapamiętana THEN
           zapamiętaj W
         wyślij potwierdzenie do O3
         IF O2 się jeszcze nie odezwał THEN
             retransmituj W do O2
        END;
    potwierdzenie z odbiornika O2:
        BEGIN
         IF  obaj sąsiedzi sprawni THEN
             zakończ synchronizację i wyjdź z pętli
        END;
    potwierdzenie z odbiornika O3:
        BEGIN
         IF  obaj sąsiedzi sprawni
             zakończ synchronizację i  wyjdź z pętli
         END
  END (* CASE *)

  IF upłynął maksymalny czas oczekiwania
        zakończ synchronizacje - wyjdź z pętli
END (* while *);

IF nie było potwierdzenia od O2 THEN
  zdiagnozuj O2
IF nie było potwierdzenia od O3 THEN
  zdiagnozuj O3
IF otrzymaliśmy wiadomość  W z więcej niż jednej stacji THEN
  zaakceptuj W
IF stacja O1 jest koordynatorem odbioru THEN
  wyślij potwierdzenie do wszystkich stacji węzła nadającego
przejdź do przetwarzania wiadomości W
```

niem jest przesyłanie potwierdzenia odebrania wiadomości W, razem z aktualnym statusem węzła O, do wszystkich stacji węzła N.

Fazy 1 i 2 operacji przekazywania wiadomości są realizowane przez stacje węzła nadawczego N, a fazy 3 i 4 – przez stacje węzła odbiorczego O.

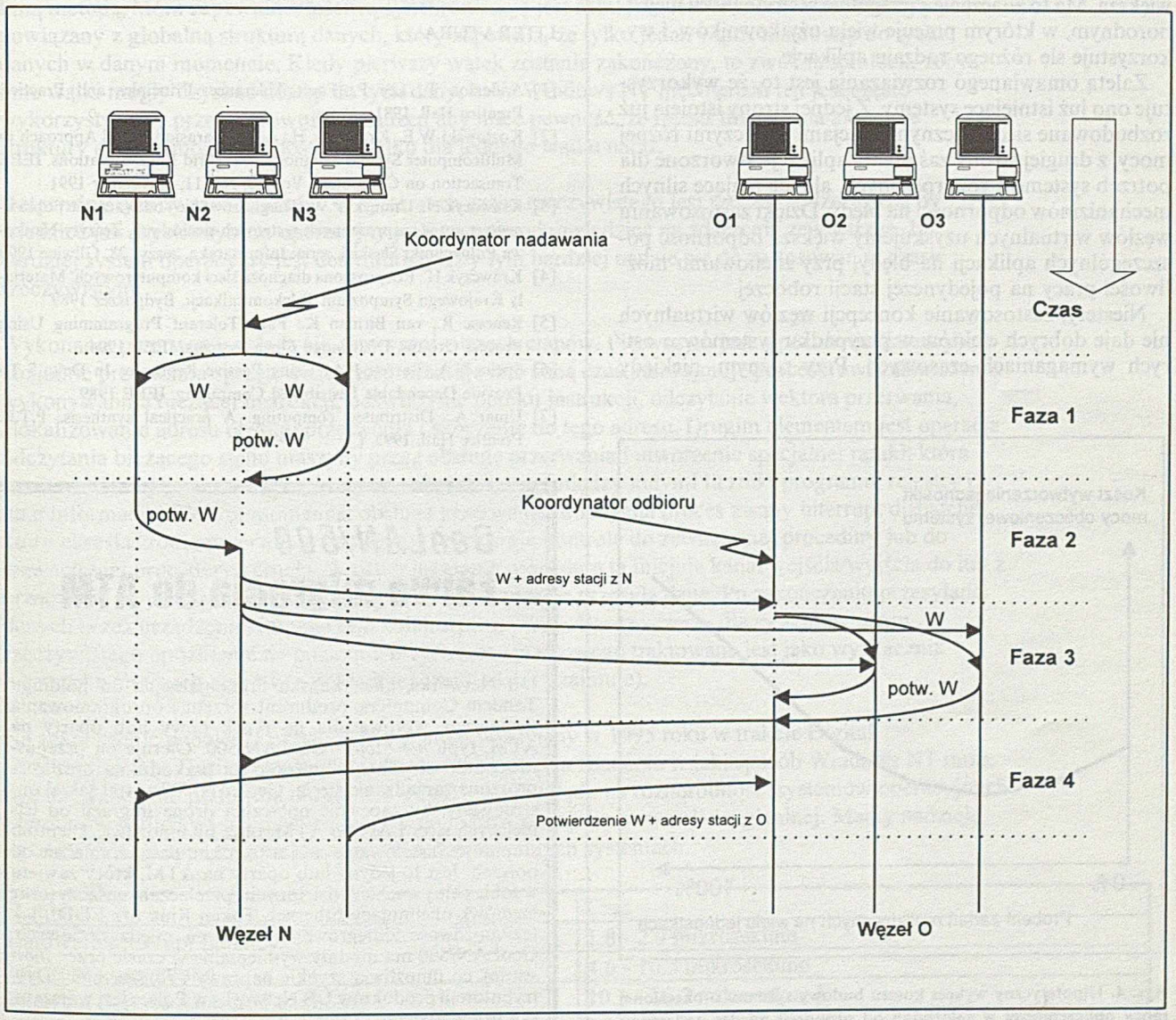
Poniżej szczegółowo przedstawiono protokół komunikacji między dwoma węzłami. Dla uproszczenia przyjęto, że każdy z węzłów składa się z trzech jednostek. Oznaczmy jednostki wchodzące w skład węzła nadawczego N przez N_1, N_2 i N_3. Węzeł N zamierza wysłać wiadomość W do węzła O, składającego się ze stacji O_1, O_2, i O_3. Przebieg fazy 1 i 2 wysyłania wiadomości przez pojedynczą stację węzła N przedstawia Algorytm 1.

Algorytm 1 zakłada, że gdy dana jednostka jest zajęta realizacją zadania Z, to wiadomości do niej są odbierane i kolejkowane tak, aby można je było przeanalizować po zakończeniu realizacji danego fragmentu zadania Z. Dzięki temu jest możliwe wzajemne zsynchronizowanie się jednostek wykonujących swoje zadania z różną prędkością.

Dalsza część operacji przesyłania wiadomości (fazy 3 i 4) odbywa się na jednostkach węzła odbiorczego O. Ilustruje to Algorytm 2.

Prezentowane algorytmy gwarantują, że wszystkie stacje węzła O otrzymają wysłaną do nich wiadomość. Wymiana komunikatów w ramach węzła wirtualnego służy nie tylko synchronizacji pracy węzłów, ale także umożliwia uaktualnienie syndromu diagnostycznego. Komunikaty te mogą bowiem zawierać nie tylko informację potrzebną do synchronizacji węzłów, ale także sygnatury cząstkowych wyników zadań realizowanych na poszczególnych węzłach, a także lokalnie określony syndrom węzła wirtualnego. Niezgodność wyników pośrednich albo brak łączności z jedną ze stacji może być podstawą do uznania jej za niesprawną i wyłączenia z dalszej pracy węzła.

Rys. 3 ilustruje przykładowe przesłanie komunikatu z węzła wirtualnego N do węzła wirtualnego O. Przedstawiono najprostszy przypadek, gdy wszystkie wiadomości wysyłane w ramach węzła wirtualnego dotarły w określonym czasie do adresatów. W wypadku braku potwierdzenia wiadomości podejrzana stacja jest diagnozowana przez pozostałe stacje przez wysyłanie komunikatów kontrolnych i zdalne uruchamianie testów. Jeśli badana stacja nie prześle prawidłowych wyników testów, to uważa się ją za uszkodzoną i zostaje ona usunięta z logicznej struktury węzła. O rekonfiguracji węzła pozo-



Rys. 3. Przesłanie komunikatu między węzłami wirtualnymi

stałe węzły informowane są podczas następnej fazy wymiany komunikatów.

* * *

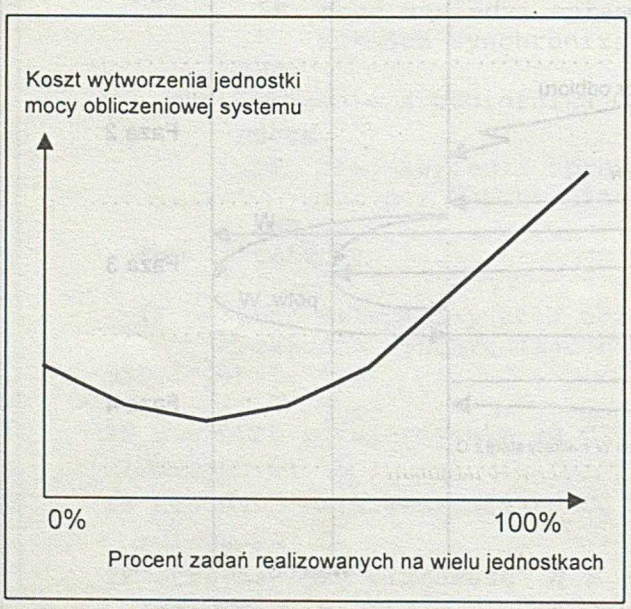
Proponowana architektura systemu odpornego na błędy opiera się na koncepcji wykonywania wielu kopii programu na wielu jednostkach jednocześnie. Klasyczne podejście do programowania *N*-wariantowego zakłada jednak wykonywanie tylko jednego procesu na każdej z dedykowanych jednostek. Zaproponowane podejście umożliwia znacznie efektywniejsze wykorzystanie istniejącej bazy sprzętowej. Stacje robocze wchodzące w skład węzła wirtualnego, stworzonego na potrzeby jednej aplikacji, mogą jednocześnie wykonywać inne zadania. W szczególności, stacja może należeć do różnych węzłów wirtualnych, działających na potrzeby różnych aplikacji. W przypadku rozproszonych baz danych, działających jako aplikacje typu klient-serwer, ta sama stacja może wykonywać program interakcyjny, będący klientem systemu, i należeć do węzła wirtualnego, spełniającego rolę serwera. W ten sposób program o małych wymaganiach niezawodnościowych (klient) wykorzystuje mniej zasobów systemu niż serwer bazy danych, którego niezawodność musi być większa. Ma to znaczenie szczególnie w środowisku niejednorodnym, w którym pracuje wielu użytkowników, i wykorzystuje się różnego rodzaju aplikacje.

Zaletą omawianego rozwiązania jest to, że wykorzystuje ono już istniejące systemy. Z jednej strony istnieją już rozbudowane sieci z licznymi stacjami roboczymi różnej mocy, z drugiej strony zaś mamy aplikacje stworzone dla potrzeb systemów rozproszonych, ale nie mające silnych mechanizmów odporności na błędy. Dzięki zastosowaniu węzłów wirtualnych uzyskujemy większą odporność poszczególnych aplikacji na błędy, przy zachowaniu możliwości pracy na pojedynczej stacji roboczej.

Niestety, zastosowanie koncepcji węzłów wirtualnych nie daje dobrych efektów w przypadku systemów o ostrych wymaganiach czasowych. Przy różnym, niekiedy znacznym, obciążeniu stacji roboczych zadaniami indywidualnymi trudno jest precyzyjnie określić czas zakończenia zadanej operacji przez węzeł wirtualny. Mechanizmy synchronizacji stacji, wchodzących w skład węzła, wprowadzają pewne opóźnienia w działaniu systemu. Podobnie w systemach, które obsługują przede wszystkim jedną aplikcję (np. zarządzającą bazą danych), przy relatywnie małej liczbie zadań indywidualnych, stosowanie węzłów wirtualnych wydaje się bezcelowe.

Rys. 4 przedstawia hipotetyczny wykres kosztu jednostki mocy systemu opartego na koncepcji węzłów wirtualnych w zależności od tego, jaki procent zadań systemu stanowią zadania o podwyższonych wymaganiach niezawodnościowych, realizowane na podstawie węzłów wirtualnych. Obliczając koszt jednostki mocy bierze się pod uwagę nie tylko koszty sprzętu potrzebnego do realizacji zadanych funkcji, ale także koszty wytworzenia oprogramowania.

Zaproponowana strategia odporności na błędy umożliwia wykorzystanie istniejącej bazy sprzętowej oraz rozwiązań programowych do budowy uniwersalnych systemów o podwyższonej odporności na błędy. Omawiane podejście wymaga dokładniejszej analizy kosztów oraz zweryfikowania eksperymentalnego.



Rys. 4. Hipotetyczny wykres kosztu budowy systemu o określonej mocy obliczeniowej w zależności od proporcji między zadaniami indywidualnymi stacji i zadaniami rozproszonymi

LITERATURA

[1] Anderson T., Lee P.: Fault Tolerance: Principles and Practice. Prentice Hall, 1981
[2] Kozłowski W.E., Krawczyk H.: A Comparasion-Based Approach to Multicomputer System Diagnosis in Hybrid Fault Situations. IEEE Transaction on Computers, Vol. 40, No. 11, November 1991
[3] Krawczyk H., Umiński P.W.: Diagnozowanie i odtwarzanie poprawnego stanu w rozproszonych systemach sieciowych, Zeszyty Naukowe Politechniki Śląskiej, seria: Informatyka, zeszyt 28, Gliwice 1995
[4] Krawczyk H.: Rozproszona diagnoza sieci komputerowych. Materiały Krajowego Sympozjum Telekomunikacji, Bydgoszcz 1989
[5] Renesse R., van Birman K.: Fault-Tolerant Programming Using Process Groups, Distributed Open Systems. IEEE 1994
[6] Speirs N.A., Barrett P.A.: Using Passive Replicates In Delta-$ To Provide Dependable Distributed Computing. IEEE 1989
[7] Umar A.: Distributed computing. A practical synthesis. P.T.R. Prentice Hall, 1993. □

GeoLAN/500
Łatwa migracja do ATM

UB Networks, amerykańska firma należąca do holdingu Tandem Computers, producent sprzętu i oprogramowania sieciowego, wprowadziła na rynek nowy hub oparty na ATM, typu *non-stop* – GeoLAN/500. Oferuje on przepustowość do 40 Gb/s, odporność na uszkodzenia oraz rozproszone zarządzanie siecią. GeoLAN/500 został tak skonstruowany, by zapewnić opłacalną drogę migracji od istniejących sieci LAN do ATM, oraz by umożliwić klientom stopniowe instalowanie sieci o szerokim paśmie, zależnie od potrzeb. Jest to jedyny hub oparty na ATM, który zawiera w sobie pełny wachlarz dzielonych i przełączanych łączy (oraz mediów), obejmujący Ethernet, Token Ring orz FDDI, jak też możliwość zintegrowania połączeń międzysieciowych. GeoLAN/500 ma moduły wymienialne w czasie pracy (*hot-swap*), co umożliwia szybkie naprawy i rozszerzenia. Dystrybutorem produktów UB Networks w Polsce jest warszawska firma Apexim. (k)

Windows NT i systemy czasu rzeczywistego (cz. 3)

Microsoft

Trzecia i ostatnia część artykułu poświęconego Windows NT i systemom czasu rzeczywistego dotyczyć będzie pozostałych dwu elementów architektury: wymagań synchronizacyjnych i deterministycznego czasu reakcji.

Wymagania synchronizacyjne. Jednym z najtrudniejszych zadań w systemach czasu rzeczywistego jest zapewnienie synchronizacji pomiędzy różnymi wątkami i procesami. Oznacza to, że w aplikacjach czasu rzeczywistego, jest niezwykle istotne w jakim czasie zachodzą określone zdarzenia. Jeśli na przykład jedna część aplikacji zakończy pracę przed tym jak druga część otrzyma najaktualniejsze dane, to może to doprowadzić do tego, że proces jaki dana aplikacja monitoruje może stać się niestabilny. Synchronizacja ma za zadanie zapewnić, że komponenty aplikacji są priorytezowane w prawidłowy sposób.

Synchronizacja kernela. Większość zadań wykonywanych w kernelu odbywa się na najwyższym programowym poziomie przerwania (dispatch level) lub powyżej niego. Zadanie kernela to przede wszystkim synchronizacja pracy na wielu procesorach, obsługa przerwań i zarządzanie systemową bazą danych. Kernel posiada również bogaty zestaw obiektów typu dispatch. Mają one za zadanie synchronizację wykonania kodu w sterownikach urządzeń i komponentach systemu Windows NT. Obiektami tymi są między innymi zegary systemowe, zdarzenia, semafory i inne.

Inną metodą, która zapewnia właściwą synchronizację jest spinlock. Jest to mechanizm blokujący powiązany z globalną strukturą danych, który zapewnia, że tylko jeden wątek ma dostęp do tych danych w danym momencie. Kiedy pierwszy wątek zostanie zakończony, to zwalnia spinlock, żeby inne wątki mogły uzyskać dostęp do tych danych. W Windows NT mechanizm ten jest często wykorzystywany przez sterowniki urządzeń aby mieć pewność, że rejestr urządzenia albo inne struktury danych mogą być dostępne tylko dla jednego wątku naraz.

Deterministyczny czas reakcji. W systemach czasu rzeczywistego jest rzeczą bardzo ważną by wiedzieć jak szybko system operacyjny jest w stanie odpowiedzieć na zdarzenia zewnętrzne. Im bardziej system operacyjny jest deterministyczny tym bardziej nadaje się do zastosowań w czasie rzeczywistym.

Wykonanie przerwania składa się zazwyczaj z trzech etapów. Pierwszym jest opóźnienie (latency) na poziomie przerwania sprzętowego. Reprezentuje ono sobą czas jaki zajmuje procesorowi zakończenie wykonywannia bieżącej instrukcji, odświeżenie potoku instrukcji, odczytanie wektora przerwania, zlokalizowanie adresu obsługi przerwania i skoczenie do tego adresu. Drugim elementem jest operacja odczytania bieżącego stanu maszyny przez obsługę przerwania i utworzenie specjalnej ramki, która zawiera stan wykonania wątku, który został przerwany (między innymi liczniki programu, rejestry i inne informacje). W tym momencie obsługa przerwania uruchamia proces zwany interrupt dispatcher, który określa źródło przerwania i następnie przekazuje kontrolę do zewnętrznej procedury lub do wewnętrznej procedury kernela. W ostatnim etapie procedura ta inicjuje kanał wejścia/wyjścia do lub z urządzenia i wykonuje inne wątki podczas gdy urządzenie przesyła dane. Po zakończeniu przesyłania danych przez urządzenie, inicjuje ono kolejne przerwanie. Bardzo często dla systemów czasu rzeczywistego opóźnienie na poziomie przerwania sprzętowego traktowane jest jako wyznacznik wszystki tych trzech etapów (a dokładniej czasu jakie to zajmuje).

Na zakończenie podajemy przykładowy pomiar wykonany w 1995 roku w trakcie Digital Communications Design Conference. Pomiar miał na celu zbadanie w jaki sposób Windows NT może być używane jako platforma dla routera TCP/IP. Ze względu na różnorodność systemów operacyjnych i terminów w nich występujących, nazwy pomiarów podajemy w wersji oryginalnej. Mamy nadzieję, że ułatwi to ich dopasowanie do odpowiedników w innych systemach.

Pomiar	Czas
Hardware Interrupt Latency	1.8 - 2.9 mikrosekund
Interrupt Dispatching	4.6 - 10.5 mikrosekund
Interrupt Service Routine Lenght	10.3 - 16.7 mikrosekund
Total Elapsed Time	16.7 - 30.1 mikrosekund

O/11/95

SYSTEMY CZASU RZECZYWISTEGO

REDAGUJE ZESPÓŁ: dr inż. Zbigniew FRYŹLEWICZ (Politechnika Wrocławska), prof. dr hab. inż. Janusz GÓRSKI (Francusko-Polska Wyższa Szkoła Nowych Technik Informatyczno-Komunikacyjnych w Poznaniu), prof. dr hab. inż. Zbigniew HUZAR (Politechnika Wrocławska), dr inż. Krzysztof SACHA (Politechnika Warszawska), prof. dr hab. inż. Tomasz SZMUC (Akademia Górniczo-Hutnicza), dr inż. Zbigniew ZIELIŃSKI (Wojskowa Akademia Techniczna), mgr inż. Zdzisław ŻURAKOWSKI (Instytut Automatyki Systemów Energetycznych we Wrocławiu) – przewodniczący zespołu.

ROK II ● LISTOPAD ● 1995 ● 11

Metody analizy i specyfikacji wymagań

Podejście strukturalne do analizy i projektowania systemów

z uwzględnieniem rozszerzeń dla systemów czasu rzeczywistego (1)

Jan Kwiatkowski

Centrum Informatyczne, Politechnika Wrocławska

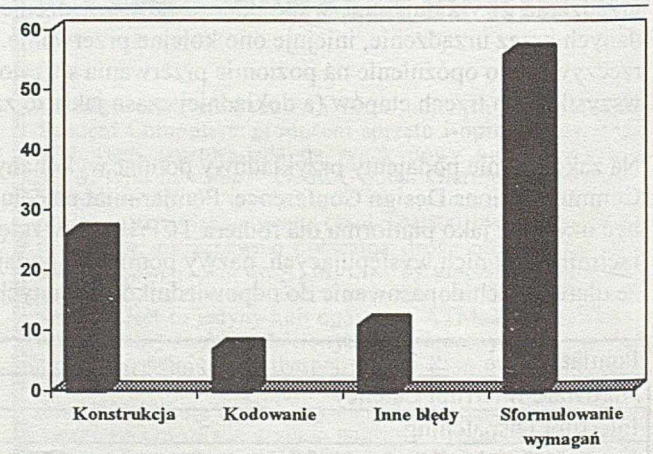
Damian Misiak, Stanisław Szejko

Zakład Zastosowań Informatyki, Politechnika Gdańska

W artykule przedstawiono strukturalne podejście do analizy i projektowania systemów. Wyjaśniono podstawowe pojęcia oraz wykorzystano w metodzie notację graficzną. Zaprezentowano rozszerzenia standardowej metody umożliwiające projektowanie systemów czasu rzeczywistego (Warda-Mellora, Hartley-Pirbhai'a). Omówiono kolejne kroki konstrukcji oprogramowania z wykorzystaniem metody projektowania strukturalnego koncentrując się na etapach analizy systemu, specyfikacji modelu oraz projektowania. Przedstawiono rozszerzenie metody umożliwiające jej wykorzystywanie do projektowania systemów współbieżnych (DARTS).

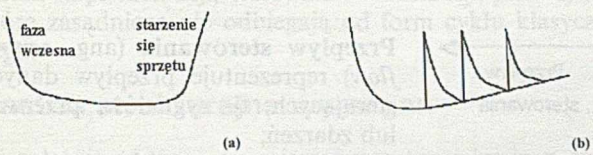
Sytuację, jaka wytworzyła się w konstrukcji systemów informatycznych w latach osiemdziesiątych [16] można scharakteryzować w następujący sposób:

- Rozwój systemów dokonuje się najczęściej według klasycznego (kaskadowego) cyklu rozwoju oprogramowania, mimo znanych jego wad, takich jak: brak płynnego przejścia od fazy analizy do fazy konstrukcji, nieuczestniczenie użytkowników w procesie projektowania, weryfikacji i atestacji dokonywanej dopiero pod koniec cyklu rozwoju. Późne wykrycie błędów istotnie zwiększa koszty ich usunięcia, a tym samym wytworzenia całego systemu. Procentowy rozkład błędów popełnianych w trakcie konstrukcji systemu ilustruje rysunek 1. Cykl klasyczny jest też źródłem znacznych kosztów ponoszonych w czasie konserwacji systemu,



Rys. 1. Procentowy rozkład błędów popełnianych przy konstrukcji systemów [24]

na skutek operowania produktem niezbyt modyfikowalnym i o niskiej jakości;
- Brak koordynacji pomiędzy projektem oprogramowania oraz konfiguracji sprzętowej systemu. Właściwie tylko faza wstępnej analizy wymagań jest wspólna, w następstwie tego integracja produktu sprzętowego i programowego jest często trudna, długotrwała oraz zazwyczaj wymaga zmian w już zrealizowanych elementach systemu. Jednocześnie w ciągu ostatnich 40 lat stosunek ceny do parametrów efektywnościowych w przypadku sprzętu poprawił się 10^{10} raza, przy zaledwie pięciokrotnym wzroście tego współczynnika dla oprogramowania [24];
- W przypadku sprzętu mamy do czynienia nie tylko ze znaczącym wzrostem mocy przy jednoczesnym spadku ceny – również kształt krzywej obrazującej liczbę uszkodzeń w funkcji czasu (rys. 2a) nie odbiega zbytnio od założeń, wykazując stabilizację po okresie konstrukcji do czasu zestarzenia się sprzętu. Ta sama krzywa dla oprogramowania (rys. 2b) wykazuje gwałtowne skoki powodowane wprowadzaniem nowych błędów w trakcie zmian wymuszonych usuwaniem istniejących błędów lub potrzebą modyfikacji oprogramowania [26]; należy zwrócić uwagę na wzrost liczby uszkodzeń, będący konsekwencją prac konserwacyjnych.



Rys. 2. Krzywe liczby uszkodzeń ujawniających się w czasie (a) – w przypadku sprzętu (b) – w przypadku oprogramowania

- Problem konstrukcji systemów sprzętowych można uznać za rozwiązany. Współcześnie sprowadza się on do doboru odpowiednich elementów katalogowych bądź do wykorzystania metod zapewniających automatyczne przejście pomiędzy opisem pożądanego zachowania układu a jego realizacją [6, 27];
- W przypadku oprogramowania wygląda to zupełnie inaczej, bo chociaż udało się przesunąć proces jego konstrukcji z dziedziny sztuki do sfery rozwiązań naukowych, to wciąż jeszcze daleko do „normalnego rzemiosła" (do metod stosowanych w innych dziedzinach przemysłu). W efekcie mamy do czynienia z niską efektywnością procesu wytwarzania oprogramowania, poważnymi problemami zapewnienia odpowiedniej jego jakości i częstym brakiem zadowolenia użytkowników, natomiast koszty wytworzenia oprogramowania są ogromne. W 1987 r. szacowano je na 140 mld USD, co przy światowej tendencji wzrostu o około 12% rocznie daje około 345 mld USD w roku 1995 [34];
- Bardzo często w konstrukcji oprogramowania mamy do czynienia z procesem, który można być nazwać „zdegenerowanym cyklem tworzenia oprogramowania": analiza wymagań „w głowie", po czym przejście do implementacji, praktycznie z pominięciem fazy projektowania. Równie częste jest oczekiwanie przyszłego użytkownika, że otrzyma bezbłędny produkt bez uprzedniego testowania go w środowisku docelowym. Podobnie niedoceniana jest faza wdrażania wytworzonego systemu;
- Wymienione zjawiska nasilają się, gdy mamy do czynienia z systemami pracującymi w czasie rzeczywistym. Połączenie środków sprzętowych, programowych i działania ludzkiego, silne powiązanie ze środowiskiem pracy, ograniczenia czasowe, ograniczona możliwość testowania w warunkach rzeczywistych zwiększają problemy konstrukcji systemów o wysokiej jakości, spełniających wymagania bezpieczeńswa i niezawodności.

Można wskazać dwie istotne tendencje, jakie pojawiły się w reakcji na wskazaną powyżej sytuację. Pierwsza z nich to **wzrost zastosowań metod modelowania systemów,** bardziej lub mniej formalnych. Powszechne stały się notacje służące do opisu struktury i przepływu danych – diagramy związków encji (ang. *Entity-Relationship Diagrams*) [7, 39], diagramy przepływu danych (ang. *Data Flow Diagrams*) [11, 26]; notacje opisu zachowania, zwykle bazujące na modelach przejść stanu (grafy, sieci Petri [28], automaty skończone i ich rozwinięcia np.: statecharts [15]); notacje opisu logiki przetwarzania, m.in. diagramy akcji, tablice decyzyjne, języki typu „structured English" [1] oraz służące opisowi struktury i zachowania systemu modele sieci procesów i modele obiektowe [37, 38]. Temu samemu celowi, tzn. opisowi oraz ocenie i weryfikacji wymagań i systemów, służyć miały metody formalnej specyfikacji, jak specyfikacje algebraiczne (abstrakcyjne typy danych) [22] lub notacja *Z* [35].

Druga z wymienionych tendencji polega **na poszukiwaniu metodologii systematycznej konstrukcji oprogramowania.** Zależnie od przyjmowanych przez nie założeń wyróżnia się *podejścia:* prototypowanie, podejście zorientowane na struktury danych, podejście strukturalne, podejście obiektowe, metodologie formalne [1, 26, 34, 41]. W każdym przypadku ułatwia to nie tylko rozwiązanie problemów technicznych i organizacyjnych, ale przede wszystkim zapewnia podniesienie jakości wytworzonego produktu, a co za tym idzie – obniżenie kosztu konserwacji systemu.

Wspólną cechę wszystkich wymienionych podejść stanowi wykorzystanie graficznych notacji do modelowania wymagań, struktury i zachowania projektowanego systemu oraz wspomaganie procesu analizy, projektowania i konstrukcji programowymi narzędziami typu CASE (*Computer Aided Systems Development Tools*) [1, 24]. Ograniczone ramy tego artykułu pozwoliły jedynie na skrótowe wskazanie w nim tych notacji i narzędzi, które wspierają podejście strukturalne i zdobyły popularność dzięki swoim praktycznym zastosowaniom.

Podejście strukturalne

Ideą podejścia strukturalnego jest hierarchiczna dekompozycja funkcjonalna z wykorzystaniem diagramów przepływu danych i sterowania. Najbardziej znanymi metodologiami tego podejścia są: Analiza Strukturalna DeMarco oraz Yourdona i Constantine, SA/SD Yourdona, SA/SD/RT Warda i Mellora [11, 40, 42]. Metoda SSADM (*Structured Systems Analysis and Design Method* [32]) stanowi popularne połączenie podejścia strukturalnego z podejściem ukierunkowanym na struktury danych.

Metodologie strukturalne mają na dzień dzisiejszy najwięcej wdrożonych, komercyjnych aplikacji, stąd modele i notacje w nich wykorzystywane oraz kolejne kroki procesu projektowania strukturalnego stanowią zasadniczy przedmiot niniejszego artykułu.

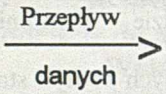
Analiza strukturalna jest podejściem zorientowanym funkcjonalnie. Cały system jest dekomponowany na zbiór wzajemnie na siebie oddziałujących, powiązanych modułów, z których każdy ma jasno zdefiniowaną funkcję. Stanowi ona zbiór narzędzi oraz technik pozwalających na modelowanie przepływu danych przez system uwzględniając różne jego aspekty. Używając tych technik tworzy się modele opisujące funkcje systemu, struktury danych, na których działa, jego zachowanie wynikające z uzależnień czasowych oraz strukturę systemu uwzględniającą powiązania pomiędzy głównymi jego komponentami.

Pierwotnie notacja dla analizy strukturalnej powstała dla analizy standardowych aplikacji przetwarzających dane [11], dopiero później wprowadzono rozszerzenia pozwalające na modelowanie systemów czasu rzeczywistego [40, 17].

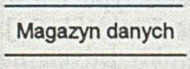
Podstawowe modele podejścia strukturalnego

Do modelowania funkcji systemu jest wykorzystywana notacja graficzna w postaci tzw. diagramów przepływu danych, obrazujących przekształcanie danych wejściowych na dane wyjściowe (wyniki) przez kolejne przekształcenia funkcjonalne. W diagramach przepływu danych są wykorzystywane następujące symbole:

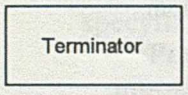
Transformacja (ang. *process*) – reprezentuje indywidualną funkcję wykonywaną przez system; funkcje przekształcają dane wejściowe w wyjściowe;



Przepływ danych (ang. *data flow*) – stanowi połączenie pomiędzy procesami i reprezentuje informacje wymagane przez proces jako jego dane wejściowe i/lub informacje, które generują jako wynik;



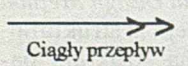
Magazyn danych – (ang. *data store*) reprezentuje zbiór danych, które system musi pamiętać przez pewien czas;



Terminator (ang. *terminator*) reprezentuje zewnętrzne zasoby spoza systemu (ludzkie, systemowe), z którymi system się komunikuje.

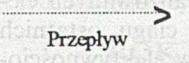
Dla opisu systemów czasu rzeczywistego przedstawiona notacja została uzupełniona o następujące symbole:

Rozszerzenia Warda – Mellora:

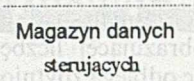


Ciągły przepływ danych (ang. *continuous data flow*) – stanowi „ciągłe" połączenie między procesami;

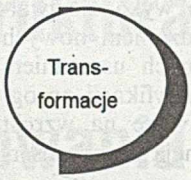
Transformacja sterująca (ang. *control process*) dokonuje przekształceń sterowania (akceptuje i generuje wyłącznie przepływ sterowania);



Przepływ sterowania (ang. *control flow*) reprezentuje przepływ sygnałów sterujących (sygnałów, przerwań, zdarzeń);

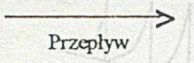


Magazyn danych sterujących (ang. *control store*) reprezentuje zbiór sygnałów sterujących pamiętanych przez system;



Transformacje (ang. *processes*), reprezentuje wielokrotne wystąpienia tego samego procesu.

Rozszerzenia Hatley-Pirbhai'a:



Przepływ sterowania (ang. *control flow*) reprezentuje przepływ danych sterujących, tj. sygnałów, przerwań lub zdarzeń;

Reguły decyzji (ang. *vertical bar*) odnośnik do specyfikacji sterowania opisującej zachowanie systemu i definiującej aktywizację procesów jako konsekwencję zdarzeń

Do modelowania danych w podejściu strukturalnym są używane diagramy związków encji, opisujące przechowywane dane w systemie na wysokim poziomie abstrakcji. Stosowana technika umożliwia, przy użyciu notacji graficznej, specyfikowanie różnych klas obiektów danych, na których operuje system oraz wzajemnych powiązań między tymi obiektami.

W przypadku modelowania zachowania systemu czasowo zależnego są używane diagramy zmian stanu (ang. *State Transition Diagrams* – STD), opisujące stany, w jakich znajduje się system oraz zdarzenia powodujące jego zmiany. Dodatkowo jest możliwe wyspecyfikowanie akcji – zdarzenia (np. aktywacja transformacji), podejmowanych jako konsekwencja wystąpienia poszczególnego zdarzenia. Yourdon [42] rozróżnia trzy klasy zdarzeń występujących w systemie: zdarzenia przychodzące do systemu z zewnątrz (od terminatorów), zdarzenia związane z upływem czasu (od „niby zegara") oraz zdarzenia powstające w systemie na skutek wykonywanych transformacji (np. przyczyną zdarzenia może być zejście poziomu zapasów magazynowych poniżej wymaganego poziomu).

Do modelowania zachowania można również stosować tablice aktywacji procesów (ang. *Process Activation Table* – PAT); w odróżnieniu od STD wskazują one, które

procesy opisane przez DFD będą uaktywniane w następstwie wystąpienia konkretnego zdarzenia.

Do specyfikacji logiki procesów stosuje się tzw. „Structured English". Stanowi on zbiór słów kluczowych wywodzących się z języka angielskiego wraz z ograniczeniami na rodzaj zdań, które razem ze zmiennymi i regułami pozwalają składać z tych zdań specyfikację procesów. Można również wykorzystać tablice decyzyjne (ang. *decision tables*). Odzwierciedlają one reguły wiążące warunki logiczne (wartości zmiennych) i akcje podejmowane przy określonych warunkach logicznych.

Dla modelowania struktury systemu jest używana zazwyczaj reprezentacja graficzna, tzw. diagramy struktury (ang. *structure chart*), opisujące hierarchiczną strukturę (dekompozycję) funkcjonalnych elementów oprogramowania. Odzwierciedlone są powiązania między produktami otrzymywanymi przy zastosowaniu podejścia strukturalnego [36].

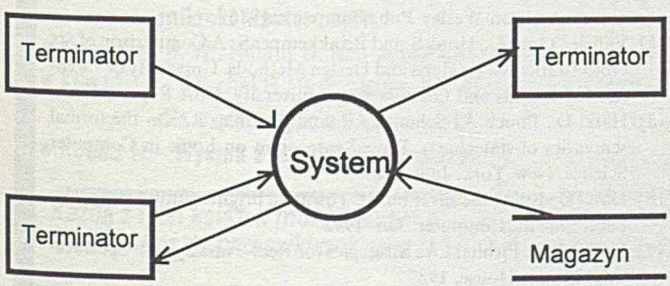
Analiza i projektowanie systemu przy pomocy podejścia strukturalnego

Ogólny cykl tworzenia systemu według metodologii strukturalnych nie odbiega od zasad przyjętych dla innych podejść systematycznych. W dalszym ciągu rozdziału są opisane kolejno fazy analizy systemu, specyfikacji modelu i projektowania. Mniej uwagi poświęcono fazom implementacji, testowania i walidacji, jako tym, które zasadniczo nie odbiegają od form cyklu klasycznego.

Analiza strukturalna wymagań

Niezależnie od tego, czy opisujemy system istniejący, czy zamierzony, zadaniem analizy wymagań jest przedstawienie modelu określającego **co** system ma robić (w odróżnieniu od fazy konstrukcji, w której określa się **jak** system ma to realizować). Przedmiotem opisu są więc: ulokowanie systemu w jego środowisku pracy, zbiór realizowanych funkcji oraz zachowanie systemu. „Klasyczne" podejście strukturalne zakłada realizację procesu analizy w pięciu krokach (jest to o tyle umowne, że czynności wykonywane w poszczególnych krokach zazębiają się wzajemnie, a ich realizacja przebiega iteracyjnie):

1. Przedstawienie *diagramu kontekstu systemu*, lokującego system w środowisku współpracujących z nim obiektów. Na tym poziomie system jest odwzorowany jako pojedyncza transformacja z oznaczonymi wszystkimi wejściowymi i wyjściowymi przepływami danych (rys. 3) do(od) terminatorów lub zewnętrznych zbiorów danych, z których system korzysta (np. przy modelowaniu „Poczty" mogłyby to być odpowiedniki skrzynek na listy).



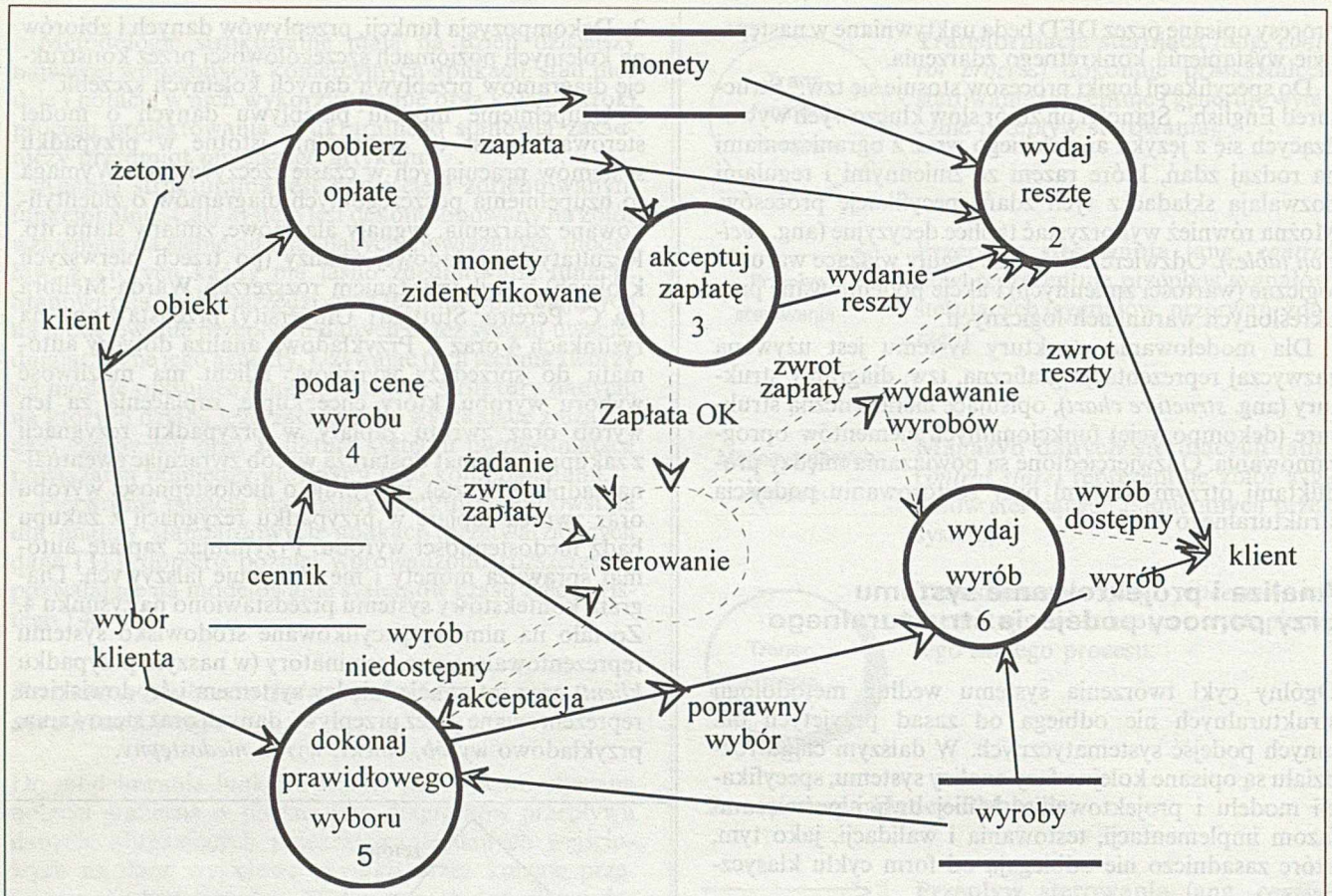
Rys. 3. Diagram kontekstu systemu

2. Dekompozycja funkcji, przepływów danych i zbiorów na kolejnych poziomach szczegółowości przez konstrukcję diagramów przepływu danych kolejnych szczebli.
3. Uzupełnienie modelu przepływu danych o model sterowania, jest to szczególnie istotne w przypadku systemów pracujących w czasie rzeczywistym. Wymaga to uzupełnienia poszczególnych diagramów o zidentyfikowane zdarzenia, sygnały alarmowe, zmiany stanu itp. Rezultaty przykładowej analizy (po trzech pierwszych krokach) z wykorzystaniem rozszerzeń Warda-Mellora (za C. Pereira, Stuttgart University) przedstawiono na rysunkach 4 oraz 5. Przykładowa analiza dotyczy automatu do sprzedaży wyrobów. Klient ma możliwość wyboru wyrobu, który chce kupić, zapłacenia za ten wyrób oraz zwrotu zapłaty w przypadku rezygnacji z zakupu. Automat dostarcza wyrób zwracając ewentualną nadpłatę (resztę), informuje o niedostępności wyrobu oraz zwraca zapłatę w przypadku rezygnacji z zakupu bądź niedostępności wyrobu. Przyjmując zapłatę automat sprawdza monety i nie akceptuje fałszywych. Diagram kontekstowy systemu przedstawiono na rysunku 4. Zostało na nim wyspecyfikowane środowisko systemu reprezentowane przez terminatory (w naszym przypadku *klient*) oraz interakcje między systemem i środowiskiem reprezentowane przez przepływy danych oraz sterowania, przykładowo *wyrób, obiekt, wyrób niedostępny*.

Rys. 4. Przykład automatu do sprzedaży wyrobów – diagram kontekstowy

Rysunek 5 przedstawia diagram poziomu 0. Stanowi on rozwinięcie (uszczegółowienie) diagramu kontekstowego. Zostało na nim wyspecyfikowane zachowanie systemu w reakcji na zewnętrzne zdarzenia. Główna transformacja *sprzedaż wyrobów* reprezentująca cały system została podzielona na sześć transformacji reprezentujących główne *moduły* funkcjonalne systemu, np.: pobranie opłaty, wydanie wyrobu, wydanie reszty. Ponadto zostały wyspecyfikowane aspekty związane z danymi (strukturami danych), przez przepływy danych wiążące transformacje (dane wymieniane pomiędzy poszczególnymi *modułami* funkcjonalnymi, np.: *zapłata, poprawny wybór*) oraz przez magazyny danych reprezentujące logiczną strukturę danych przechowywanych przez system, np.: *wyroby, cennik*. Wyróżniono również transformację sterującą akceptującą i generującą takie sygnały sterujące, jak: zwrot zapłaty, żądanie zwrotu zapłaty, wyrób niedostępny.

4. Opis zachowania systemu, mający zazwyczaj postać *specyfikacji sterowania*. Zależnie od notacji, przyjmuje on



Rys. 5. Przykład automatu do sprzedaży wyrobów; diagram poziomu 0

najczęściej formę diagramów przejść stanu lub tablic aktywacji procesów.

5. Opis przetwarzania dokonywanego przez procesy najniższych poziomów. Opis ten może być zrobiony w języku naturalnym, może zawierać równania bądź tabele, zwykle jednak dokonuje się go przy pomocy pewnego języka specyfikacji, jak Structured English. Można również wykorzystać tzw. tablice decyzyjne.

Niektóre z metodologii zakładają dekompozycję modelu systemu na *model środowiska i model zachowania*. W metodzie SA/SD przyjmuje się, że opis środowiska obejmuje diagram kontekstu systemu oraz *listę zdarzeń zewnętrznych*, na które system winien odpowiadać. W celu identyfikacji tych zdarzeń jest zalecane posłużenie się diagramem związków encji, opisującym powiązania podstawowych obiektów diagramu kontekstu systemu. Natomiast opis zachowania stanowi opis transformacji dokonywanych w odpowiedzi na poszczególne zdarzenia z listy. W zależności od przyjętej notacji można posłużyć się diagramami przepływu danych bądź diagramami transformacji (czyli diagramami przepływu danych rozszerzonymi o przepływ i transformacje sygnałów sterujących) oraz przyjętym językiem minispecyfikacji opisujących informację i sposób jej przekształcenia.

Dokończenie w następnym numerze

LITERATURA

[1] Avison D.E. and Fitzgerald G.: Information Systems Development: Methodologies, Techniques and Tools. Blackwell Scientific Publications, 1988

[2] Benders L.P.M. and Stevens M.P.J.: Tasl level behavioral hardware description. Proc. Microprocessing and Microprogramming'91 Conf. North-Holland 1991

[3] Booch G.: Object-oriented development. IEEE Trans. on Software Eng., Vol. SE-12, No. 2., February 1986

[4] Broy M. and Wirsing M. (eds): Selected Papers on CIP Project. Lecture Notes on Comp. Sci., Springer-Verlag 1990

[5] Broy M. and Wirsing M. (eds): Methods of Programming. Selected Papers on the CIP-Project. LNCS No. 544, Springer-Verlag, 1991

[6] Camposano R.: From Behaviour To Structure High-Level Synthesis. IEEE Design and Test of Computers, October 1990

[7] Chen P.P.: The Entity-Relationship Model – Towards a Unified View of Data. ACM Transactions on Database Systems, Vol. 1, No. 1, March 1976

[8] Coad P. and Yourdon E.: Object-Oriented Analysis. Yourdon Press, 1991

[9] Coad P. and Yourdon E.: Object-Oriented Design. Yourdon Press, 1991

[10] Cohen B., Harwood W.T. and Jackson M.I.: The Specification of Complex Systems. Addison-Wesley Pub. Comp. 1986

[11] DeMarco T.: Structured Analysis: Systems Specifications. Prentice-Hall, 1979

[12] Dunn R.: Software Quality Assurance, Prentice-Hall, 1990

[13] Gehrani N. and McGettrick A. (eds): Software Specification Techniques. Addison-Wesley Pub. Company, 1986

[14] Van den Goor G., Hong S. and Brinkkemper S.: A Comparison of Six Object-Oriented Analysis and Design Methods. University of Twente, Netherlands and Georgia state University, USA Report, 1992

[15] Harel D., Pnueli A., Schmidt J.P. and Sherman R.: On the formal semantics of statecharts. Proc. Symposium on Logic in Computer Science, New York, June 1987

[16] Harel D.: Biting the silver bullet. Toward a brighter future for system development. Computer, Jan. 1992

[17] Hatley D.J., Pirbhai I.A.: Strategies for Real-Time System Specification. Dorset House, 1987

[18] IEEE Software No. 7, Vol. 2. Special Issue on Software Metrics, Mar 1990

[19] Jones C.B.: Systematic Software Development Using VDM. Prentice Hall, 1986
[20] Lano K. and Haughton H. (eds): Oriented Specification Case Studies. Prentice Hall, 1994
[21] Lipsett R., Schaefer C.F. and Ussery C.: VHDL: Hardware Description and Design. KLUWER Academic Publishers, 1990
[22] Liskov B. and Guttag J.: Abstraction and Specification in Program Development. MIT Press, McGraw-Hill Book Company, 1986
[23] Łyskawa H., Misiak D., Piechówka M.: Narzędzia CASE – charakterystyka i zastosowania. Konferencja „Informatyka na wyższych uczelniach dla gospodarki narodowej", t. I, s. 170. Gdańsk, 1994
[24] Parkinson J.: Making CASE Work. NCC Blackwell Ltd., 1991
[25] Peng Z., Fagerström J., Kuchciński K.: A Unified Approach to Evaluation and Design of Hardware/Software Systems. 13th ICSE Workshop, Austin, Texas, May 1991
[26] Pressman R.S.: Software Engineering. A Practicioner's Approach. McGraw-Hill, Inc., 1992
[27] Rammig F.J.: Approaching system level design. Invited paper to EURO-VHDL Conference, Sept. 1991, Stockholm
[28] Reisig W.: Sieci Petriego. WNT, Warszawa, 1988
[29] Rumbaugh J., Blaha M. Premerlani W., Eddy S. and Lorensen W.: Object-oriented Modeling and Design. Prentice-Hall 1991
[30] Sannella D.: A survey of formal software development methods. Integrated Systems Factory Study report, 1989
[31] Stepney S., Barden R. and Cooper D. A survey of object orientation in Z. Software Engin. Journal, March 1992
[32] Skidmore S., Farmer R. and Mills G.: SSADM Vesion 4. Models and Methods. NCC Blackwell Ltd., 1992
[33] Shlaer S. and Mellor S.: Object Lifecycles. Modeling in the World of States. Yourdon Press, 1992
[34] Sommerville I.: Software Engineering. Addison-Wesley, 1992
[35] Spivey J.M. The Z Notation. A Reference Manual. Prentice-Hall, 1992
[36] Skidmore S. and Wroe B.: Introducing Systems Design. NCC Blackwell 1990
[37] Szejko S.: Design models in computer sytems. Education in Computers, Vol. 4, No. 1, 1992
[38] Szejko S.: System desing methods. Education on Computers Vol. 5, No. 2, 1993
[39] Tsichritzis D.C. and Lochovsky F.H.: Data Models. Prentice Hall Software Series, 1982
[40] Ward P.T. and Mellor S.J.: Structured Development for Real-Time Systems. Yourdon Press, 1985
[41] Wrycza St.: Współczesne metodyki tworzenia systemów informatycznych zarządzania. Raport nr 45 Programu CPBR.5-636/89, Gdańsk 1989
[42] Yourdon E.: Modern Structured Analysis. Prentice-Hall, 1989. □

ERRATA

We wrześniowym numerze INFORMATYKI do artykułu pp. Janusza Górskiego i Bartosza Nowickiego *„Obiektowe podejście do modelowania i analizy komputerowych systemów sterowania"* wkradły się błędy, które poniżej prostujemy, przepraszając za zaistniałą sytuację Autorów i Czytelników.

- Strona 15, szpalta 1, linia 20 - powinno być: **na rysunkach 3, 4, 5, 6 i 7.**
- Rys. 1: - w pierwszym prostokącie od lewej strony, w linii trzech innych prostokątów (środek rysunku) powinna być nazwa: **TempSensor.**
- Rys. 6:
  - łuk łączący prostokąty COLLECTINGGAS I VENTILATION powinien być skierowany w przeciwną stronę;
  - etykieta na łuku od prostokąta C do NOTENOUGHTOIGN powinna być ujęta w nawiasy prostokątne **[CONCENTRATION <= C_IGN]**;
  - etykieta na łuku od prostokąta C do ENOUGHTOIGN powinna być ujęta w nawiasy prostokątne **[CONCENTRATION>C_IGN].**
- Rys. 9: (tworzony na podstawie rys. 6)
  - uwagi jak przy rys. 6 oraz
  - od dolnego prostokąta VENTILATION do górnego prostokąta powinien być łuk.
- Strona 19, szpalta 1 od góry - powinno być ~~NOGAS-TOOMUCHGAS~~.

Wydawnictwa Naukowo-Techniczne
00-048 Warszawa, Mazowiecka 2/4

POLECAJĄ! POLECAJĄ! POLECAJĄ! POLECAJĄ!

- Berglin S., Howes C., Jones S. - **Word 6 dla Windows (przewodnik obrazkowy)** -25,00 zł
- Booth D. - **Symphony 2** - 4,00 zł
- Botto F. - **Paradox** - 5,00 zł
- Dunn S., Bermant Ch., Berst J. - **Windows 3.1 - 101 wskazówek i trików** - 5,00 zł
- Ferbache D. - **Patologia wirusów** - 12,00 zł
- Harris W. - **Bazy danych nie tylko dla ludzi biznesu** - 12,00 zł
- Hertz J., Krogh A., Palmer R. G. - **Wstęp do teorii obliczeń neuronowych** - 12,00 zł
- Holland R. - **Testowanie i diagnostyka systemów mikrokomputerowych** - 9,00 zł
- Kernighan B. W., Ritchie D. M. - **Język ANSI C** -15,00 zł
- Klin M. Ch., Pöschel R., Rosenbaum K. - **Algebra stosowana dla matematyków i informatyków** - 6,00 zł
- Koblitz N. - **Wykład z teorii liczb i kryptografii** - 15,00 zł
- Kozielski S., Szczerbiński Z. - **Komputery równoległe. Architektura, elementy programowania** - 7,50 zł
- Kyle J. - **Zarządzanie pamięcią w Windows** -12,50 zł
- Leyland V. - **EDI. Elektroniczna wymiana dokumentacji** - 18,00 zł
- Nance B. - **Windows for Workgroups** - 21,00 zł
- Robling Denning D. E. - **Kryptografia i ochrona danych** - 13,50 zł
- Smith R. - **PostScript** - 25,00 zł
- Software W. - **SQL. Język relacyjnych baz danych** - 14,00 zł
- Southerton A., Perkins E. C., Jr. - **Słownik poleceń systemów Unix i X** - 43,00 zł
- Spence J. - **Excel 5** - 17,80 zł
- Stroustrup B. - **Język C++** - 34,00 zł
- Weiss Z., Gruźlewski T. - **Programowanie współbieżne i rozproszone w przykładach i zadaniach** - 9,00 zł
- Wood L. - **Poczta elektroniczna. Modemy i programy telekomunikacyjne** - 16,00 zł

Każdą z tych książek można nabyć za zaliczeniem pocztowym. Przy zamówieniu o wartości przekraczającej 25,00 zł koszty przesyłki pokrywa WNT, a dołączenie kuponu upoważnia do uzyskania 10% rabatu.

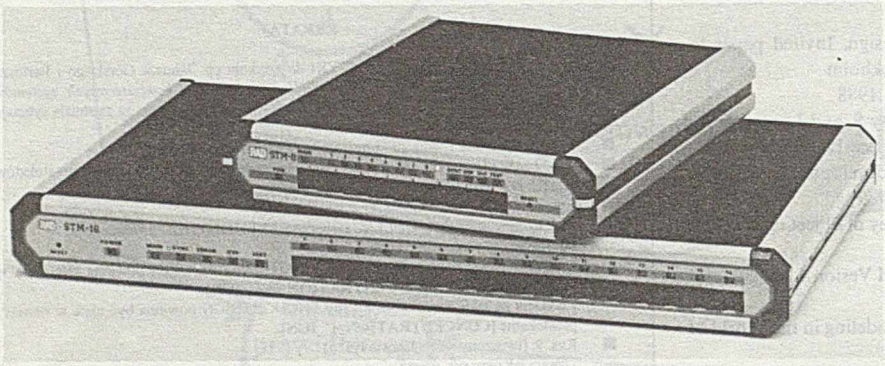
Nasz adres: Dział Handlowy WNT, skr. poczt. 359, 00-950 Warszawa, tel./fax 26-82-93

KUPON Informatyka 3/95

0/21/95

informacje

Nowe multipleksery firmy RAD



Firma RAD Data Communications ogłosiła wprowadzenie na rynek nowych modeli multiplekserów statystycznych rodziny STM. Są to STM-16, multiplekser o szesnastu kanałach łącznej przepustowości do 38,4 kbps, pomyślany do umiarkowanych obciążeń oraz STM-8/HS i STM-16/HS, urządzenia odpowiednio ośmio- i szesnastokanałowe o przepustowości 64 kbps, do obciążeń dużych.

Rodzina STM obejmuje obecnie dwadzieścia cztery modele: dwu-, cztero-, ośmio-, szesnasto- i dwudziestoczterokanałowych urządzeń, z czego cztery ostatnie również w wersji szybkiej (HS). Umożliwiają one transmisję asynchroniczną od 75 do 19200 bps na kanał. Niezależna konfiguracja kontroli przepływu dla każdego kanału pozwala użytkownikom podłączać różne rodzaje terminali do tego samego multipleksera. Rozbudowane są korekcja błędów i automatyczna retransmisja. **(PB)**

Coś dla oka:

Graph Expo 95

W dniach 8–11 października odbyła się w Chicago impreza o nazwie Graph Expo – targi idei, produktów i usług, które kształtują współczesny świat graficznego przekazu. Niemal jedna trzecia terenu wystawy była przeznaczona na eksponaty z dziedziny elektronicznego przygotowania publikacji – od graficznego projektu do odwzorowania obrazu z komputera na matrycę. Cała dostępna przestrzeń tego segmentu wystawienniczego została wyprzedana już wiosną. Resztę zajęli wystawcy reprezentujący silnie obecnie skomputeryzowane technologie drukarskie, druk cyfrowy i inne pokrewne dziedziny.

Na początku lata liczba zgłoszonych wystawców sięgała pięciuset. Spodziewano się ponad pięćdziesiąt tysięcy zwiedzających. Teren wystawy zajął ponad 34 000 m^2, co stawia Graph Expo w rzędzie największych wydarzeń handlowych USA roku 1995. **(PB)**

Usługi interakcyjne Microsoftu

Microsoft podpisał porozumienie z ponad 60 firmami programistycznymi, które zobowiązały się do podjęcia prac nad stworzeniem pakietu ułatwiającego zarządzanie małą firmą, jak również innych aplikacji i usług dla powstającej sieci Microsoft – MSN (*The Microsoft Network*). Microsoft planuje rozbudowanie w przyszłym roku działu sieci oraz towarzyszącego mu zestawu usług serwisowych i programów, w tym m.in. kolumn i czasopism ogłoszeniowych, oprogramowania wspomagającego zarządzanie, służb konsultingowych i doradców prawnych. Wśród zaproponowanych przez Content Providers rozwiązań i usług informacyjnych warto wymienić możliwość dostępu do bazy danych zawierającej informacje związane z małym biznesem (*Minority Business Network*), zestaw do interakcyjnego informowania o możliwościach kredytowych (*Dun & Bradstreet*), forum on-line wymiany doświadczeń pracowników branży restauracyjnej i noclegowej (*PalCom*) i wiele innych. Założenia strategiczne Microsoft Network – nowej usługi interakcyjnej, zostały przedstawione przez Microsoft w listopadzie ub.r. MSN miał od początku w założeniu pomóc twórcom interakcyjnego oprogramowania usług w rzeczywistym wykorzystaniu potencjału, jaki zawiera technologia sieciowa. Dostęp do sieci będzie oferowany wszystkim użytkownikom Windows 95 jako opcjonalna odpłatna usługa. Mogą oni korzystać z zasobów o zasięgu światowym płacąc jedynie za lokalne połączenie telefoniczne. (k)

„Chomik" zamiast myszy

Nowe urządzenie do sterowania kursorem amerykańskiej firmy Cirque – GlidePoint, które otrzymało pieszczotliwe imię „chomik" – oferuje na naszym rynku warszawska firma ATTI. Zaletą „chomika" jest to, iż nie ma ruchomych części mechanicznych, co sprawia, że jest bardzo trwały. Zbliżenie palca do powierzchni sensorycznej powoduje zmianę pojemności między warstwami wstęg przewodzących, determinując ruch jeszcze przed dotknięciem „chomika". Dostępna jest też na naszym rynku klawiatura firmy Cirque z wbudowanym „chomikiem".

WINDOWS '95

na notebooku LTE 5000

Firma Compaq, która ma tytuł *Microsoft Lead Systems Partner for Windows 95*, zaprezentowała na światowej premierze Windows 95 przeszło 80 swoich produktów, w tym nowy notebook Compaq LTE 5000. Jest on używany przez sprzedawców Microsoftu w USA do demonstrowania Windows 95 i Windows NT. LTE 5000 to nowa seria modularnych notebooków wyposażonych w 64-bitowe procesory Pentium, grafikę PCI local bus i możliwości rozszerzenia pamięci operacyjnej RAM do 72 MB. Architektura modularna umożliwia klientom wbudowanie napędu CD-ROM, do 2,7 GB pojemności dyskowej, dołączenia dwóch baterii lub napędu dyskietek. Funkcjonalność multimedialną zapewnia adapter wideo MPEG i TV w połączeniu z cyfrowym stereofonicznym audio i głośnikami stereo montowanymi przy monitorze. Absolutną nowością jest modularna MultiBay Expansion Base, która współpracuje z notebookiem za pomocą tych samych wzajemnie wymienialnych urządzeń, obejmujących pojemność dyskową do 5,4 GB i oferującą dwa dodatkowe gniazda PCMCIA, zintegrowany Ethernet i głośniki stereo.

ALM - ułatwienie dla programistów

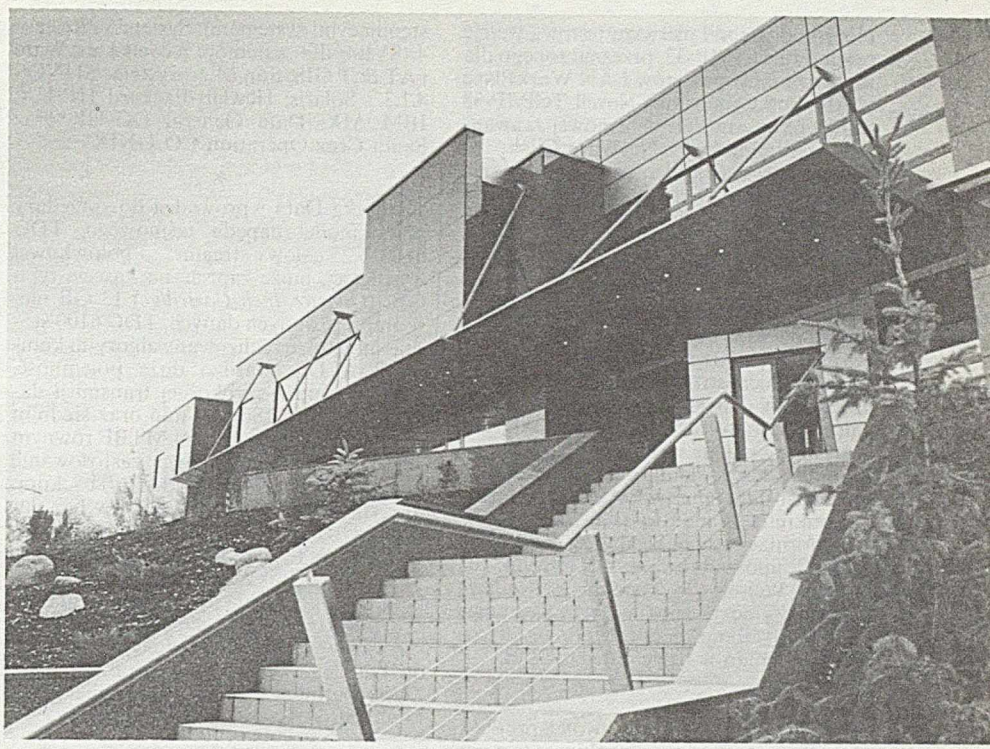
Novell zakończył prace nad modułem ALM (*AppWare Loadable Module*) obsługującym formanty VBX (*Visual Basic Controls*). Moduł ten umożliwia współpracę między językiem Visual Basic, a oprogramowaniem AppWare, które jest oparte na technologii modułowej wysokiego poziomu (5GL) i służy tworzeniu aplikacji dla komputerów Macintosh i systemu Windows. To zintegrowane środowisko umożliwia programistom szybkie i efektywne tworzenie nowych aplikacji za pomocą oprogramowania AppWare. Programiści używający modułu VBX ALM mogą z tysięcy istniejących bibliotek VBX wybierać formanty i wprowadzać je do środowiska AppWare. Po dodaniu formantów VBX do projektu można z nich korzystać w identyczny sposób, jak z innych wizualnych elementów dostępnych w środowisku AppWare, np. obiektów tekstowych, tabel lub przycisków. Moduł VBX ALM jest bezpłatnie dostępny przez następujące kanały rozpowszechniania:

- AppWare: CompuServe – Go AppWare
- Novell Bulletin Board Service (BBS) – 801-221-5197
- AppWare Internet Support – AW-Sup @Novell.com
- AppWare Internet Mailing List – Appware-info @adeva.org
- AppWare Home Page – WWW – http://www.novell.com/ServSupp/dev-sup/appware.

Ponadto moduł VBX ALM będzie dołączony do następnej wersji pakietu AppWare, która ma być gotowa w końcu 1995 r. (k)

CA-CLIPPER 5.3
z narzędziami graficznymi

Firma Computer Associates International oferuje nową, udoskonaloną wersję najpopularniejszego systemu do tworzenia aplikacji baz danych dla DOS – CA-Clipper 5.3. Nowością jest możliwość używania przy programowaniu narzędzi graficznych działających w środowisku Windows. Zintegrowane środowisko programisty pracujące w Windows jest bardzo podobne do środowiska oferowanego przez CA-Visual Objects i zawiera m.in. graficzne edytory do tworzenia menu i formularzy, DBServer, browsery aplikacji, modułów, procedur i błędów, wbudowany debugger do testowania DOS-owych aplikacji pod Windows, wbudowane repozytorium z automatyczną funkcją MAKE. CA-Clipper 5.3 ma też wiele innych nowości, które pozwalają programiście na pracę pod DOS-em lub przejście do środowiska Windows i wykorzystanie wszystkich dostępnych narzędzi graficznych. Dotychczasowi użytkownicy mogą nabyć nową wersję jako upgrade, jest też ona dostępna na CD-ROM. Polskim dystrybutorem CA jest firma SoftPoint.



Trzeci, siódmy i piąty
HECTOR zmienia siedzibę i strategię

Istniejąca od 1984 r. firma HECTOR S.A. przeniosła się do nowej siedziby na warszawskim Żoliborzu (ul. Gwiaździsta 19), do funkcjonalnego i pięknego budynku (patrz zdjęcie), który został zgłoszony do konkursu „Mister Warszawy". Ta rodzinna firma (właścicielami jest małżeństwo pp. Łukszów) zajęła w 1994 r. siódme miejsce w rankingu największych firm komputerowych na polskim rynku, trzecie wśród producentów komputerów i piąte wśród firm o największym kapitale i funduszach.

Nabywcami stu tysięcy wyprodukowanych przez HECTORA komputerów są m.in. Sejm, Senat, banki, wojsko, szpitale, uczelnie i urzędy administracji państwowej. Obecnie produkcja stanowi jednak tylko 38% obrotów firmy (z tego 20% na rynkach krajów b. ZSRR).

HECTOR, doceniając zmiany jakościowe na naszym rynku komputerowym, zmienia teraz strategię: zacieśnia współpracę z Digitalem (oferuje komputery tej firmy w poważnych przetargach), interesuje się specjalistycznymi systemami 64-bitowymi opartymi na procesorach RISC, zamierza coraz bardziej specjalizować się w integrowaniu systemów (współpracuje w tej dziedzinie z firmą Inwar). Chce też uzyskać na swoje komputery i usługi normy ISO. (k)

Rozbudowanie oprogramowania Transcend

3Com poinformowała, iż dodaje do oprogramowania *Transcend Enterprise Management*, które jest aplikacją zintegrowanego zarządzania sieciami, technologię zdalnego monitorowania RMON (*Remote Monitoring*). Technologia RMON obniży koszty zarządzania siecią i zmniejszy jej złożoność. Firma zapewnia, że rozbudowa ta nie wpłynie na cenę. Obecna wersja RMON pochodzi od Token Ring RMON. Przyszli i obecnie użytkownicy, nie ponosząc dodatkowych kosztów, będą korzystać z usprawniających pracę funkcji RMON, w które będą wyposażane koncentratory LinkBuilder MSH Ethernet i wieżowe koncentratory LinkBuilder II Ethernet. W przypadku, gdy RMON nie będzie mógł być zastosowany w zewnętrznych koncentratorach 3Com, które są przyłączone do koncentratorów przełączających LANPlex 6000, RMON Probe może być zdalnie przyłączany, aby sprawdzić każdy port w koncentratorach LANPlex połączonych siecią światłowodową FDDI. Transcend RMON jest w pełni zgody ze standardami RFC1271 i współpracuje z dodatkowymi programami 3Com, takimi jak AutoCalibre, ułatwiając obsługę zdalnego monitorowania RMON.

Novell udostępnił pierwszą testową wersję programu TCP/IP-32, przeznaczonego dla użytkowników pakietów LAN WorkPlace i LAN WorkGroup firmy Novell. TCP/IP-32 stanowi kluczowy element nowej zaawansowanej architektury NetWare Client 32, która zapewnia bardzo szybkie działanie sieci TCP/IP przy radykalnym zmniejszeniu zużycia pamięci konwencjonalnej w komputerach osobistych.

*

Firma Alcatel Cabling System opracowała program CMS (*Cabling Management Software*), który ułatwia właściwe zarządzanie okablowaniem strukturalnym służącym do przesyłania danych, obrazu, głosu i sygnałów kontrolnych. CMS bazuje na graficznej prezentacji systemu okablowania. Administrator systemu może uzyskać natychmiast dane identyfikujące użytkownika konkretnego stanowiska pracy. Wszystkie informacje są pokazane w strukturze „okien", uporządkowane i z łatwym dostępem. Wejście do systemu jest zabezpieczone (kluczykiem i hasłem), aby uniemożliwić osobom nieuprawnionym dokonywanie zmian w bazie danych CMS.

*

3Com dołączył do rodziny swoich produktów ethernetowy serwer zdalnego dostępu – AccessBuilder 2000. Umożliwia on transmisję z prędkością do 115,2 Kbps, jest oparty na procesorze RISC i960. Zdalni użytkownicy mogą łączyć się z serwerem w systemach DOS, Windows, Macintosh lub stacji roboczych UNIX oraz uzyskać dostęp do aplikacji LAN pracujących na serwerach TCP/IP, Novell NetWare, Microsoft LAN Manager, Windows for Workgroups, Windows NT oraz Apple Talk.

*

Novell oferuje program uaktualniający arkusz kalkulacyjny Quattro Pro, który jest częścią pakietu biurowego PerfectOffice. Umożliwia on użytkownikom arkusza Lotus 1-2-3 łagodne przejście do Quattro Pro. Uaktualniona wersja arkusza kalkulacyjnego Quattro Pro dokonuje także konwersji plików i stylów utworzonych w programie Microsoft Excel 5.0.

*

American Power Conversion rozpoczął produkcję dwóch nowych wysokopojemnościowych zasilaczy Back-UPS Pro, które stanowią uzupełnienie serii UPS-ów line-interactive, wyposażonych w system Automatycznej Regulacji Napięcia (AVR). Zasilacze Back-UPS Pro 1000 i 1400 VA zapewniają podtrzymanie bateryjne i ochronę przed przepięciami komputerów oraz innego sprzętu elektronicznego. Są wyposażone w system inteligentnego zarządzania baterią, który jest przeznaczony do monitorowania wydajności, żywotności oraz niezawodności baterii. Nowe zasilacze są przeznaczone dla wielu systemów komputerowych, telekomunikacyjnych, kas fiskalnych wymagających długotrwałego podtrzymywania bateryjnego.

*

Elektroniczny system przesyłania wiadomości GroupWise 4.1 – strategiczny element pakietu sieciowego PerfectOffice Novella, z którego korzysta już ponad 3,5 mln użytkowników, jest dostępny dla siedmiu znakowych odmian systemu MotifUNIX. Obecnie GroupWise współpracuje z następującymi systemami: System V, Release 4 (SVR4) dla systemów Novell UnixWare i AT & T GIS; Sun Microsystems SUNOS 4.1.3 i Solaris; Hewlett-Packard HP-UX; IBM AIX; Data General DG-UX oraz Santa Cruz Operation CO UNIX.

*

Tandberg Data wprowadza do sprzedaży nowy model napędu taśmowego. TDC 6100, 5,25 calowy streamer, o połówkowej wysokości, może zapisać na kasecie typu QIC (*Quarter Inch Cartridge*) 13 GB nieskompresowanych danych. TDC 6100 stosuje sprzętowo realizowany algorytm kompresji ALDC. Bardzo duża pojemność, w połączeniu z szybkością transmisji danych od 22 do 180 MB/min oraz średnim czasem międzyawaryjnym MTBF równym 300 tys. godz., umożliwia zastosowanie napędu m.in. do zapisu wideo lub backupu dużych, rozproszonych systemów baz danych. Urządzenie pracuje na platformie Windows NT, OS/2 Warp, LAN Manager, SCO Unix, UnixWare, Solaris i in. Tandberg Data podpisała już umowy z kilkoma firmami, które włączą napędy do pakietów softwarowych swojej produkcji, służących do kontroli sporządzania kopii (backup). Dystrybutorem produktów Tandberg Data w Polsce jest Sof-tronik Polska.

*

Autodesk wprowadził do wersji 12 Auto-CAD nowe sterowniki umożliwiające bezpośrednie łączenie obiektów-rysunków sporządzonych w formacie CAD z bazą danych Oracle for Microsoft Windows 3.1. Stwarza to nowe możliwości analizowania i sporządzania raportów. Sterowniki typu ASI (*AutoCAD SQL Interface*) zostały zaprojektowane z myślą o użytkownikach 12 wersji AutoCAD – mających moduł Auto-CAD Data Extension (ADE). ADE ułatwia zadawanie pytań, dotyczących projektów sporządzonych w formacie CAD, łączących dane graficzne i tekstowe. Auto-desk zapowiada pojawienie się sterownika typu ASI dla wersji 13 AutoCAD-a pozwalającego na współpracę z Oracle v.7 oraz czterech innych sterowników ASI dla wersji 12: Dbase III i Dbase IV dla DOS i Windows 3.1. Nowe sterowniki będą nieodpłatnie udostępniane wszystkim zarejestrowanym użytkownikom wersji 12 AutoCAD.

*

Europejski Ośrodek Ekspertyz należący do International Data Corporation opublikował listę firm, które w pierwszych miesiącach 1995 r. sprzedały najwięcej komputerów przenośnych. Pierwsze miejsce zajmuje Compaq (19,1%), na drugim jest Toshiba (16,4%), na trzecim IBM (9,3%). Dalsze miejsca zajmują: Apple (6,8%), AST (3,9%), Dell (3,8%), SNI (2,8%), Hewlett-Packard (2,7%), Vobis (2,3%) i Escom (1,9%).

*

Digital Equipment Corporation po raz pierwszy od pięciu lat zanotowała zyski: czwarty kwartał roku finansowego 1995, który skończył się 30 czerwca, przyniósł firmie zysk w wysokości 160 mln USD, podczas gdy w analogicznym okresie 1994 r. odnotowano straty w tej samej wysokości. O 32% wzrosły w ciągu roku obroty w zakresie systemów AlphaGeneration. W kwietniu br. DEC zaprezentował najpotężniejsze produkowane na skalę przemysłową wieloprocesorowe serwery przeznaczone do obsługi dużych aplikacji.

Zaprosili nas...

Centrum Promocji Informatyki na I Ogólnopolską Konferencję promocyjną „INTERNET w Polsce". W trakcie jej trwania zaprezentowano pierwszą w Polsce transmisję obrad konferencji w sieci Internet (WWW i kanał IRC), przedstawiono pokaz tworzenia serwisów WWW za pomocą oprogramowania WEB MAGIC firmy SILICON GRAPHICS oraz poinformowano zebranych o zbliżającym się INFO-FESTIWALU'95, który obędzie się w Krakowie. Redakcja nasza objęła patronat prasowo-promocyjny nad tym festiwalem.

Altkom Matrix do odwiedzenia swojego stoiska w trakcie wystawy SOFTARG'95, odbywającej się w Katowicach w dniach 26–29 września 1995 r.

Digital Equipment Polska na premierę najszybszego w Polsce wieloprocesorowego superserwera AlphaServer 8400 Turbolaser.

Zarząd Targów Warszawskich Biura Reklamy SA na kolejne spotkanie Klubu Prasy Komputerowej. Gościem na tym spotkaniu był Pan Janusz Maszkiewicz, dyrektor Departamentu Informatyki Urzędu Rady Ministrów.

OPTIMUS SA na prezentcję własnych osiągnięć z okazji „Dni otwartych OPTIMUSA 16–17 września 1995 r." W trakcie spotkania zaprezentowano osiągnięcia firm współpracujących z OPTIMUSEM.

Zarząd Federacji Stowarzyszeń Naukowo-Technicznych NOT na konferencję związaną z mającą się odbyć VIII Ogólnopolską Konferencją NOTEX'95 na temat „Proeksportowa polityka – warunkiem wzrostu gospodarczego kraju".

SAS Institute na dwudniową II Ogólnopolską Konferencję Użytkowników Systemu SAS – SASForum'95. Konferencji towarzyszyły warsztaty szkoleniowe oraz ekspozycje strategicznych partnerów SAS Institute.

JM Rektor prof. dr hab. Bohdan Gruchman i Senat Akademii Ekonomicznej w Poznaniu na uroczystą inaugurację 70 roku akademickiego, otwarcie biblioteki głównej AE oraz inaugurację działalności laboratorium zintegrowanych systemów informatycznych Katedry Informatyki Ekonomicznej. Wykład inauguracyjny „Zintegrowane systemy informatyczne w gospodarce" wygłosił prof. dr hab. inż. Witold Abramowicz.

Minister Łączności oraz **szef Wojsk Łączności i Informatyki Sztabu Generalnego WP** na IV Międzynarodową Konferencję i Wystawę pn. „Systemy łączności i informatyki na potrzeby obrony i bezpieczeństwa RP".

Oficyna Informatyczna oraz **Fundacja Kultury Informatycznej** na kolejne czwartkowe spotkanie poświęcone systemom wspomagającym wytwarzanie aplikacji – rodzinie produktów ACUCOBOL'85.

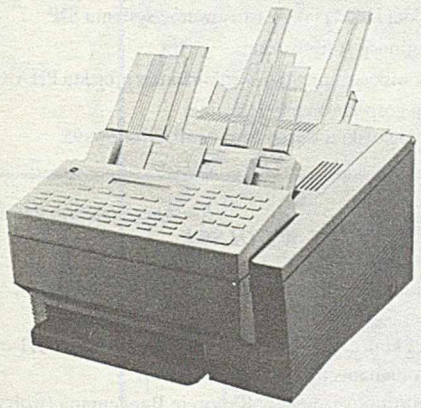
TECMATH Polska Sp. z o.o., z siedzibą w Warszawie, oraz **TECMATH GmbH**, z siedzibą w Kaiserslautern w Niemczech, na prezentację programów komputerowych GIDAS i TimeEdit, będących nowoczesnymi narzędziami służącymi do szeroko rozumianej analizy danych pomiarowych. **(t)**

Faksy też coraz inteligentniejsze...

Wielofunkcyjny Minoltafax 3700

Standardowo 1 MB pamięci, może mieć 3 MB Podwójny dostęp

Minolta wprowadziła na rynek wielofunkcyjny faks – Minoltafax 3700. Umożliwia on korzystanie z nowej cyfrowej technologii realizacji połączeń. Opcjonalny interfejs RS 232C i odpowiednie oprogramowanie może realizować połączenia z komputerem wyposażonym w kartę faksową i skaner. Minoltafax 3700 ma standardowo 1 MB pamięci z możliwością rozszerzenia do 3 MB. Daje to użytkownikowi możliwość dostosowania faksu praktycznie do dowolnej aplikacji. Powiązanie systemu laserowego wydruku z unikatowym systemem Fine Micro-Toning Minolta umożliwia uzyskanie czystego wydruku bezpośrednio na papierze A4.



Minoltafax 3700 ma kilka użytecznych funkcji, m.in. Mailbox TX/RX, dzięki której jest możliwe wysłanie dokumentu do innego faksu wyposażonego w „skrzynkę pocztową" i przechowywanie go w celu późniejszego wydrukowania. Skrzynkę uzupełnia hasło, które uniemożliwia przesłanie dokumentu tajnego do osoby nieupoważnionej. Urządzenie ma także funkcję podwójnego dostępu, która pozwala na transmitowanie zapamiętanego wcześniej dokumentu i jednoczesne zapamiętywanie następnego. Można także jednocześnie odbierać przychodzący dokument i drukować zapamiętany wcześniej.

Caviar AC31600

Napęd dyskowy 1,6 GB

Specjalnie do współpracy z Pentium

Western Digital rozpoczął produkcję trzyipółcalowego napędu dyskowego 1,6 GB w standardzie Enhanced IDE i o wydajności 5200 obrotów na minutę. Napędy Caviar AC31600 zostały zaprojektowane w celu zapewnienia pojemności i wydajności niezbędnych dla komputerów Pentium PC. Obecne są już w komputerach Globalyst AT&T, systemach Della i Gateway 2000 Inc.

Napęd ten umożliwia uzyskanie maksymalnych parametrów w środowisku szyny lokalnej przez użycie Mode 4 PIO o przepustowości 16,6 MB/s (programowanego wejścia-wyjścia) oraz Mode 2DMA 16,6 MB/s (bezpośredni dostęp do pamięci) ze słowem wielokrotnym. Caviar AC31600 zawiera CacheFlow4 do zaawansowanego, adaptacyjnego buforowania zapisu i odczytu. Dystrybutorem Western Digital w Polsce jest Computer 2000.

DEC

DEC MLS+v.3.1

Digital Equipment Corporation wprowadził do sprzedaży DEC MLS+ w wersji 3.1 – najnowszą wersję systemu operacyjnego UNIX(R) o podwyższonym poziomie bezpieczeństwa. System ten oparty na Digital UNIX, zawiera rozwiązania gwarantujące bezpieczną pracę w oknach w trybie B1 dla stacji roboczych (w trybie CMW – *Compartment Mode Workstation*) oraz w środowisku sieciowym dla 64-bitowej rodziny Alpha Digital. Wielość zabezpieczeń jednej platformy, zagwarantowana przez DEC MLS+ zmniejsza koszt początkowy bezpiecznego przetwarzania danych w przedsiębiorstwch. System uwzględnia m.in. niezbędną kontrolę dostępu typu Bell-LaPadula, gdzie użytkownicy nie mogą odczytać danych zapisanych na wyższym poziomie dostępu ani zapisać danych na niższym poziomie dostępu.

Nowa opcja

Oracle 7

Oracle Corp. wprowadziła na rynek nową opcję bazy Oracle 7 o nazwie VLM (*Very Large Memory*). Jest ona przeznaczona dla systemów UNIX-owych o architekturze 64-bitowej. Pierwszą platformą, wykorzystującą tę technologię, jest nowy 64-bitowy system z serii AlphaServer 8000 DEC-a.

Nowy produkt Oracle'a pozwala systemom klient-serwer obsługiwać bardzo duże bazy danych, wykorzystywane przez takie aplikacje, jak programy wspomagania decyzji, magazynowania danych, aktualizowane na bieżąco światowe systemy informacji geograficznych (GIS) czy też telewizja „na żądanie". Tego typu aplikacje tradycyjnie pracują na maszynach typu *mainframe*.

Thosiba

Dyski do notebooków

Toshiba zapowiada wprowadzenie do sprzedaży nowych dysków 2,5": wysokopojemny model MK2720FC/FB-1.35GB oraz super płaski (12,7 mm wysokości) MK1926FCV/FBV-815 MB. Oba modele są przeznaczone do notebooków i komputerów przenośnych. Zastosowano w nich najnowsze technologie kodowania informacji PRML (*Partial Response, Maximum Likelihood*). Obydwa modele dysków są dostępne w dwóch odmianach. Konstrukcja dysków MK1926F jest oparta na trzech talerzach, o wysokości napędu 12,5 mm i wadze 150 g; prędkość obrotowa 4200 obr./min, średni czas dostępu 13 m/s. Bufor danych wyposażono w pamięć o pojemności 128 KB (*Read Ahead Cache*). Dysk 1.35 GB jest zbudowany z pięciu talerzy, waży 220 g. Pozostałe parametry są takie same, jak w poprzednim modelu. Dystrybutorem produktów Toshiba DPD w Polsce jest firma INITEL-Servis.

Sporządzona pod kierunkiem Jacka Dongarry

Lista TOP 500

Przodownik: Silicon Graphics

W czerwcowym wydaniu listy TOP 500, sporządzonej w USA przez niezależnych ekspertów, pod przewodnictwem Jacka Dongarry, przedstawiającej 500 producentów superkomputerów legitymujących się największą liczbą zainstalowanych systemów oraz 500 najsilniejszych komputerów świata, na pierwszym miejscu znalazła się firma **Silicon Graphics** ze 128 instalacjami na świecie. Dotychczasowy lider – Cray Research – spadł na drugie miejsce (125 instalacji). Na trzecim miejscu jest IBM (72 instalacje).

Na liście 500 najsilniejszych komputerów świata pojawił się po raz pierwszy polski system; na 467 miejscu znalazł się **Power Challenge Silicon Graphics z Poznańskiego Centrum Superkomputerowo-Sieciowego.** Centrum TOP 500 odgrywa podobną rolę w świecie komputerów, jak lista 500 największych firm „Fortune" w świecie biznesu.

Uwzględniono UNIX

AutoCAD dla Alpha Systems

Cray spada na drugie miejsce

Autodesk, który wprowadził na rynek 13 wersję pakietu AutoCAD poinformował, że uwzględnione będą w niej cztery platformy UNIX i Windows NT, pracujące z wykorzystaniem rozwiązania Alpha Systems. Na rynku są już dostępne platformy dla SunSPARCstation, Hewlett-Packard 9000 Seria 799 i Windows NT obsługiwane za pomocą serwerów Alpha Systems. W ciągu najbliższych miesięcy Autodesk ma wprowadzić oprogramowanie dla stacji roboczych Silicon Graphics oraz platform IBM RISC System/6000.

AutoCAD Release 13 for UNIX oferuje te same możliwości zaspokajania zróżnicowanych potrzeb użytkowników, co pozostałe rozwiązania z 13 wersji AutoCAD-u.

Opracowała: Krystyna KARWICKA-RYCHLEWICZ

INFO FESTIWAL '95
Kraków, 22-25.XI.1995 r.

Microsoft Windows 95
Premiera Polskiej Wersji

Zapraszamy Wystawców do udziału w ekspozycjach tematycznych w Muzeum Narodowym:

INFORMATYKA DLA NAUKI
INTERNET W POLSCE
SYSTEMY INFORMACJI PRZESTRZENNEJ
INFORMATYZACJA WOJSKA I POLICJI
TELEKOMUNIKACJA
MULTIMEDIA
INFORMATYKA DLA BIZNESU

Firmy uczestniczące w wystawie mają możliwość prezentacji swych ofert także w programie konferencji.

odrębna ekspozycja w Galerii TPSP:

Microsoft Windows 95
Premiera Polskiej Wersji

APLIKACJE DLA WINDOWS'95

PLATFORMY SPRZĘTOWE

INFO-FESTIWAL'95 poza częścią wystawową, to przede wszystkim liczna konferencje i seminaria. Proponujemy uczestnictwo w konferencjach:

21-23.XI	*ZAAWANSOWANE TECHNOLOGIE INFORMATYCZNE W NAUCE POLSKIEJ*
21-22.XI	*INFORMATYKA W MEDIACH - Specjalna propozycja dla dziennikarzy prasowych, radiowych i telewizyjnych*
23.XI	*POUFNOŚĆ I BEZPIECZEŃSTWO DANYCH OSOBOWYCH W BAZACH DANYCH*
23-24.XI	*SYSTEMY INFORMACJI PRZESTRZENNEJ*
24-25.XI	*INTERNET W POLSCE*
24-25.XI	*TELEINFORMATYKA W WOJSKU I POLICJI*
30.XI-1.XII	*TELEINFORMATYKA W TURYSTYCE I HOTELARSTWIE*

Poza tym zapraszamy na bezpłatne seminaria organizowane 23.XI.95 przez firmy:

IMPOL - *Nowoczesne systemy przekazywania danych*

Przedstawicielstwo Deutsche Messe AG
- Sympozjum promocyjne nowej wystawy CeBIT HOME

TECHMEX International
- Bazy danych firmy Microsoft w strukturze klient-serwer

W dniach 24 lub 25.XI.1995 promocyjne konferencje

Microsoft Windows 95
Premiera Polskiej Wersji

INFORMACJE I ZGŁOSZENIA

Centrum Promocji Informatyki
tel. (022) 621.76.26, 693.59.46
fax (022) 693.59.49

Wybrane programy konferencji:

21-23 listopada '95 — *Koszt uczestnictwa: 360 zł*

ZAAWANSOWANE TECHNOLOGIE INFORMATYCZNE W NAUCE POLSKIEJ

W programie m.in.:
- Prezentacje ośrodków Komputerów Dużej Mocy
- Prezentacje Firm
- Modelowanie i obliczenia w naukach biomolekularnych
- Modelowanie i obliczenia w fizyce i chemii
- Nowe rozwiązania informatyczne dla nauk technicznych i środowiskowych
- Prezentacje Firm
- Polskie rozwiązania software'owe i hardware'owe
- Premierowa prezentacja polskiej wersji Microsoft Windows95

23-24 listopada '95 — *Koszt uczestnictwa: 250 zł*

SYSTEMY INFORMACJI PRZESTRZENNEJ

W programie m.in.:
- Stan informacji przestrzennej w Polsce
- Standaryzacja - miejsce i zakres w systemach przestrzennych
- Krajowy Rejestr Urzędowy Podziału Terytorialnego Kraju
- Badania naukowe dotyczące systemów informacji przestrzennej
- Mapy numeryczne (cyfrowe): dla systemów SIG (GIS), dla systemów SIT (LIS)
- Gdzie jesteśmy w systemie informacji przestrzennej -dyskusja panelowa.
- Rola systemu katastralnego w systemie finansowym państwa
- Efekty zastosowania informacji o terenie w zarządzaniu gminą
- Założenia systemu informacyjnego dla potrzeb rozwoju komunalnego
- Koncepcja informacji przestrzennej dla woj. ostrołęckiego
- Bank Informacji Miejskiej (BIM) jako zastosowanie systemu SIP
- Osiedla miejskie w informacji statystycznej
- System zarządzania środowiskiem - przedstawienie efektów projektu PHARE
- Program informatyzacji gospodarki przestrzennej
- Premierowa prezentacja polskiej wersji Microsoft Windows95

24 - 25 listopada '95 — *Koszt uczestnictwa: 250 zł*

INTERNET W POLSCE

W programie m.in.:
- Od poczty elektronicznej do telekonferencji - 25 lat INTERNET
- Jak dołączyć się do sieci
- Uzyskiwanie zezwoleń i koncesji na świadczenie usług INTERNET-owych
- Bezpieczeństwo i ochrona danych w sieci INTERNET
- Wizja społeczeństwa informatycznego w Raporcie Bangemana (wpływ infostrady na społeczeństwo i gospodarkę)
- Telekonferencje - prezentacja zastosowań
- Premierowa prezentacja polskiej wersji Microsoft Windows95
- Polscy sprzedawcy usług w sieci INTERNET
- INTERNET dla biznesu
- Jak prowadzić biznes za pomocą INTERNET-u
- Elektroniczne zakupy
- Media w INTERNET (prasa, radio i telewizja)
- Nauka i rozrywka w INTERNET

24-25 listopada '95 — *Koszt uczestnictwa: 240 zł*

TELEINFORMATYKA W WOJSKU I POLICJI

W programie m.in.:
- Podstawowe problemy bezpieczeństwa systemów teleinformatycznych w wojsku i policji
- Poufność i bezpieczeństwo danych osobowych w komputerowych bazach danych - projekty regulacji prawnych
- Ochrona sieci informatycznych
- Bezpieczne systemy operacyjne
- Polityka bezpieczeństwa danych
- Przestępstwa komputerowe
- Systemy informatycznego wspomagania szkolenia
- Symulatory działań bojowych i operacyjnych
- Systemy wizualizacji
- Zintegrowane systemy obsługi logistycznej policji
- Mapy numeryczne w zastosowaniach wojskowych i policyjnych
- Analiza ryzyka
- Procedury przetargowe - roczne doświadczenia z funkcjonowania ustawy o dostawach publicznych
- Premierowa prezentacja polskiej wersji Microsoft Windows95

ZAPRASZAMY NA INFO-FESTIWAL

Perfect-Ekspert AITECH

20 LAT DOŚWIADCZEŃ WDROŻENIOWYCH

Wysoka jakość

„Polifarb" Cieszyn S.A.
Zakłady Rowerowe „ROMET"
Krywałd „ERG" S.A.
Elektrownia Rybnik Huta Łaziska
Fabryka Osi Napędowych - Radomsko
Zakłady Tłuszczowe „BIELMAR"
Elektromontaż 3 Huta Bankowa
Zakłady Metalurgiczne „Trzebinia"
Walcownia Metali „Dziedzice" Huta Szkła „Kunice"
POLAM Kostuchna Zakłady Gumowe Górnictwa
Huta 1-go Maja IBM PC IBM PC IBM PC IBM PC
ODRA 1305

Wysoka jakość

AS/400 AS/400 AS/400 AS/400

1995 Zakres funkcjonalny

Sieciowy i zintegrowany system informatyczny, głównie dla dużych firm produkcyjnych i handlowych. Przystosowany do elektronicznej wymiany dokumentów (EDI - zamówienia, faktury, WZ, itp.), wielowalutowości, denominacji złotego oraz rachunkowości zarządczej. Wyposażony w system ochrony danych generowanych wewnątrz i na zewnątrz przedsiębiorstwa. Współpracujący z systemem ekspertowym ISAF (Inteligentny System Analiz Finansowych) wspomagający **podejmowanie decyzji finansowych na średnich i wyższych szczeblach zarządzania.**

AMANDVELL

System Informacji Kierownictwa	Inteligentny System Analiz Finansowych	System Finansowo-Kosztowy	System Wyposażenie
System Obrotu Towarowego i Fakturowania	System Obrotu Materiałami i Ewidencji Zakupu	Rozliczenie Zakupu Materiałów	System Środki Trwałe
System Kadry - Płace	System Analiz Marketingowych	System Technicznego Przygotowania Planowania i Rozliczania Produkcji	EDI TMW Ultra - Perfect

1987 Zakres funkcjonalny

Oferujemy kompleksowe usługi w zakresie:
- doradztwa organizacyjnego, merytorycznego, technicznego, consulting,
- organizacji, projektowania i wykonawstwa innych systemów informatycznych,
- wdrażania, konserwacji i modernizacji systemów informatycznych,
- projektowania, wykonawstwa dowolnego typu sieci komputerowych, z uwzględnieniem już istniejących rozwiązań,
- dostawa sprzętu komputerowego i sieciowego,
- szkoleń użytkowników w zakresie operatora i administratora systemu AS/400, jak również obsługi pakietów programowych PC Support/400 lub Client Access/400 przeznaczonych do współpracy z PC.

JUNISOFTEX

JUNISOFTEX

44-100 Gliwice, ul. Kościuszki 1 c, box 507
Tel./fax (0-32) 31-51-31, 31-57-00
38-80-52, 38-80-54

1195

Cena 4,50 zł (45 000 zł

KARTA PKO VISA CLASSIC TWOJĄ DEWIZĄ

Na całym świecie zastąpi Ci gotówkę, czek i kartę telefoniczną.

Przy użyciu karty **PKO VISA Classic** zarówno w Polsce jak i za granicą możesz płacić swoimi pieniędzmi zgromadzonymi w PKO BP bez konieczności posługiwania się gotówką lub czekiem.

Karta **PKO VISA Classic** umożliwia pobieranie gotówki w bankomatach i kasach banków na całym świecie.

Chcąc płacić kartą **PKO VISA Classic** za granicami Polski nie trzeba blokować środków na rachunku dewizowym.

Bank raz w miesiącu rozlicza wszystkie transakcje dokonane przy użyciu karty, obciążając w kolejności określonej przez klienta banku jego rachunki w PKO BP.

Kartę **PKO VISA Classic** można uzyskać już po roku posiadania konta osobistego w Powszechnej Kasie Oszczędności.

PKO

0127/95